

Universitat de Lleida

TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Mireia Martinez Morell

Titulació: Grau en Enginyeria Electrònica Industrial i Automàtica

Títol de Treball Final de Grau: **Implementing a discrete control system to command a prototype-bar position using a brushless motor propeller**

Director/a: Tomàs Pallejà Cabré i Juan Antonio Garriga Castillo

Presentació

Mes: Abril

Any: 2019

IMPLEMENTING A DISCRETE CONTROL SYSTEM TO COMMAND A PROTOTYPE-BAR POSITION USING A BRUSHLESS MOTOR PROPELLER

Author: Mireia Martinez Morell

Supervisor Professor: Tomàs Pallejà Cabré
Second Supervisor Professor: Joan Antonio Garriga Castillo

ABSTRACT

The design and construction of a motor-propeller-rectangular bar system prototype is proposed in this project. Components selection, system integration and verification are described, as well as the control strategy used for the system.

The motor-propeller-bar prototype consists of a rectangular bar fixed on one of its ends using a hinge, allowing the vertical movement on the other edge (one degree of freedom). A propeller allocated at the end of the bar creates the thrust force which generates the movement. The propeller is powered by a brushless motor for which its rotational speed controls the thrust force exerted to the complete system. The position of the bar is given by a distance measuring sensor that provides the distance between a set reference point and the actual point where the bar is positioned.

The system control was carried out via an Arduino Board, a controller was implemented after analysing the data by using the MatLab Software environment. After determining the transfer function of the system with MatLab by using the tools called System Identification and PID Tuner, a PI close-loop control was defined and the proportional and integration constants were transferred to the controller code for the Arduino. These variables had to be recalculated using MatLab Simulink in order to increase the efficiency of the control. The desired position of the prototype could then be set in using the Arduino interface.

The aim of this project is to summarize the knowledge needed to design and construct the prototype of the motor-propeller-rectangular bar system and to implement its control system. In addition, the results of the real system are compared to the system's model created in MatLab Simulink.

1. Index of contents

| | |
|---|----|
| ABSTRACT | 4 |
| 1. Index of contents | 5 |
| 2. Index of tables | 7 |
| 3. Index of figures | 7 |
| 4. Index of images | 8 |
| 5. Notation | 9 |
| 6. State of art and object | 10 |
| 7. Introduction | 11 |
| 7.1. Content | 11 |
| 7.2. Gantt Diagram | 12 |
| 8. Components of the system | 13 |
| 8.1. Detailed information of the plant | 13 |
| 8.2. Brushless Motor | 16 |
| 8.2.1. Characteristics | 17 |
| 8.2.2. Operating the system | 17 |
| 8.2.3. Chosen Motor: 2206-2150KV Multistar | 18 |
| 8.3. Electronic Speed Controller or ESC | 19 |
| 8.3.1. Chosen ESC: Afro ESC 30A | 20 |
| 8.4. Power Unit | 20 |
| 8.4.1. Chosen Power Unit: Agilent 6573A DC Power Series | 21 |
| 8.5. Distance measuring sensor | 21 |
| 8.5.1. Chosen distance measuring sensor: GP2Y0A41SK0F | 22 |
| 8.6. Accelerometer | 24 |
| 8.6.1. Used accelerometer: MMA7341LC | 24 |
| 8.7. Arduino | 26 |
| 8.7.1. Arduino Due | 27 |
| 8.7.2. Arduino Mega 2560 | 28 |
| 8.7.3. Comparison between Arduino Due and Arduino Mega 2560 | 29 |
| 8.7.4. Chosen Arduino: Arduino Due | 29 |
| 8.8. MatLab | 30 |
| 8.8.1. Version of MatLab: R2018a and R2013a | 30 |
| 8.8.2. Simulink and PID Tuner | 30 |
| 9. Control system | 31 |
| 9.1. P, PI and PID controllers | 31 |
| 9.1.1. P Controller | 32 |
| 9.1.2. PID Controller | 32 |
| 9.1.3. PI Controller | 33 |

| | |
|---|----|
| 9.1.4. Anti-windup | 33 |
| 9.1.5. Saturator | 34 |
| 9.2. Digitalization of a controller | 34 |
| 10. Behaviour of the system | 35 |
| 10.1. Arduino | 35 |
| 10.1.1. Test 1: Arduino Due | 36 |
| 10.1.2. Test 2: Arduino Mega | 36 |
| 10.1.3. Test 3: Arduino Due with an amplifying circuit | 36 |
| 10.2. Behaviour of the ESC | 39 |
| 10.2.1. Test 1: Working Range | 39 |
| 10.2.2. Test 2: Reprogramming | 40 |
| 10.3. Accelerometer | 42 |
| 10.3.1. Test 1: Performing the previous tests using the output of the accelerometer | 43 |
| 10.3.2. Test 2: Analysing the output of the accelerometer | 44 |
| 10.3.3. Replacement of the sensor | 45 |
| 10.4. Distance Measuring Sensor | 46 |
| 10.4.1. Test 1: Performing the previous tests using the output of the distance sensor | 47 |
| 10.4.2. Test 2: Analysing the output of the distance sensor | 48 |
| 10.5. Mathematical model: Linearization of the transfer function | 49 |
| 11. Implemented controller | 52 |
| 11.1. Characteristics of the system | 52 |
| 11.2. System designed | 52 |
| 11.3. Improved controller | 56 |
| 12. Conclusions | 63 |
| 12.1. Personal assessment | 63 |
| 12.2. Future steps on this project | 63 |
| 13. References | 64 |
| Annex I. Datasheet of the distance measuring sensor | 64 |
| Annex II. Datasheet of the accelerometer | 64 |
| Annex III. Arduino Due code to read the output of the sensors and generate the PWM signal | 64 |
| Annex IV. Arduino Mega code to read the output of the sensors and generate the PWM signal | 64 |
| Annex V. Datasheet of the brushless motor | 64 |
| Annex VI. Datasheet of the Afro ESC 30A | 64 |

2. Index of tables

| | |
|--|----|
| Table 1. Sharp sensors with different operating ranges..... | 22 |
| Table 2. Comparison between Arduino Due and Arduino Mega..... | 29 |
| Table 3. Accelerometer sensor pins and functions..... | 42 |
| Table 4. Results for the plant of the PID control after analysing the data from the accelerometer .. | 45 |
| Table 5. Distance sensor pins and functions..... | 46 |
| Table 6. Comparison between systems | 61 |

3. Index of figures

| | |
|---|----|
| Figure 1. Plant of the system, in red the static part and in green the mobile part | 13 |
| Figure 2. Plant of the system zooming the distance sensor | 14 |
| Figure 3. Plant of the system zooming the moving part..... | 15 |
| Figure 4. Schematic connection between the ESC-Arduino-Motor..... | 16 |
| Figure 5. Brushed and brushless Motor Schema (analog.com, 2019) | 17 |
| Figure 6. Schema of the outrunner and inrunner motors (analog.com, 2019) | 18 |
| Figure 7. PWM signal for different percentages of the duty cycle values (arduino.cc, 2019)..... | 19 |
| Figure 8. Sensor working principle..... | 21 |
| Figure 9. Chosen sensor operating range | 23 |
| Figure 10. Distance measuring sensor detecting an object range..... | 23 |
| Figure 11. Orientation of the 3 axes of an accelerometer..... | 24 |
| Figure 12. Plant of the system indicating the distance from the accelerometer to the hinge and motor | 25 |
| Figure 13. Parts of an Arduino DUE | 27 |
| Figure 14. Parts of an Arduino MEGA 2560 | 28 |
| Figure 15. Open-loop system general structure | 31 |
| Figure 16. Closed-loop general structure | 31 |
| Figure 17. Structure of a Proportional control..... | 32 |
| Figure 18. Structure of a Proportional-Integral-Derivative (PID) control | 32 |
| Figure 19. Structure of a Proportional-Integral (PI) control | 33 |
| Figure 20. Structure for an anti-windup applied to a PI controller (in blue the saturator and in purple the anti-windup)..... | 34 |
| Figure 21. Stable (green areas, '1' and '2') and unstable zones (red areas, '3', '4' and '5') | 35 |
| Figure 22. Amplifier Circuit (top) and amplifier circuit connections (below) | 37 |
| Figure 23. Generation of the signal flowchart | 38 |
| Figure 24. Oscilloscope - Signal generated by the Arduino | 38 |
| Figure 25. Reading sensor's output flowchart | 39 |
| Figure 26. BLHeli interface | 41 |
| Figure 27. Interface for the AFRO ESC of the BLHeli software..... | 41 |
| Figure 28. Choosing interface of the BLHeli software | 42 |
| Figure 29. Circuit implemented Motor-Accelerometer-Arduino | 43 |
| Figure 30. Output from the accelerometer, in green the analysed part | 44 |
| Figure 31. Considered data from the accelerometer..... | 44 |
| Figure 32. Data from the accelerometer being analysed | 45 |
| Figure 33. Data taken from the distance sensor..... | 47 |
| Figure 34. Data of the distance sensor with the reference point | 48 |
| Figure 35. General scheme of the system | 52 |
| Figure 36. System implemented with the MatLab values | 52 |
| Figure 37. PID Tuner Tool interface with the imported data..... | 53 |
| Figure 38. PID Tuner response to the system implemented | 53 |

| | |
|---|----|
| Figure 39. Output given using MatLab values (at the top is the step introduced, below is the output of the system) | 55 |
| Figure 40. Implemented PID system in the real prototype..... | 55 |
| Figure 41. Simulink response for the real implemented system (at the top is the step introduced, below is the output of the system)..... | 56 |
| Figure 42. Data recorded for a constant duty with different reference distances 85 V [top], 150V [middle], 100V [bottom] | 57 |
| Figure 43. PID Tuner output to implement a PID controller for the open-loop system | 59 |
| Figure 44. PID Tuner output to implement a PI controller for the open-loop system | 59 |
| Figure 45. Improved PID controller schema on Simulink..... | 60 |
| Figure 46. Final results for the improved PI controller (at the top is the step introduced, below is the output of the system) | 60 |
| Figure 47. Comparison between the system's output on Simulink (From top to bottom: the input step, the output from the MatLab system, output of the implemented System and the improved system only simulated) | 61 |
| Figure 48. Comparison superimposed between the system's output on Simulink (in orange the input step, in purple the output from the MatLab system, in pink output of the implemented System and in blue the improved system only simulated) | 62 |

4. Index of images

| | |
|---|----|
| Photo 1. Entire system motor-propeller-bar | 13 |
| Photo 2. Distance sensor with the added cardboard | 14 |
| Photo 3. Rotational part of the system..... | 15 |
| Photo 4. Brushless Motor (hobbyking.com, 2019) | 18 |
| Photo 5. Afro ESC 30A (hobbyking.com, 2019)..... | 20 |
| Photo 6. Power Unit used to perform the tests (keysight.com, 2019) | 21 |
| Photo 7. Aluminium bar with the accelerometer (with the wires connected on the left, without wires on the right)..... | 25 |
| Photo 8. Chosen accelerometer..... | 25 |
| Photo 9. Arduino UNO | 26 |
| Photo 10. Arduino DUE | 27 |
| Photo 11. Arduino MEGA 2560..... | 28 |
| Photo 12. Signal generated by the Arduino DUE (3.3 V) | 36 |
| Photo 13. Signal generated by the Arduino DUE and the amplifier circuit (5 V)..... | 37 |
| Photo 14. AfroESC USB programming Tool..... | 40 |
| Photo 15. SimonK online programming tool | 40 |
| Photo 16. Plant of the system indicating the accelerometer | 43 |
| Photo 17. Zoom in the distance sensor and condenser..... | 46 |
| Photo 18. Plant of the system indicating the distance sensor..... | 47 |
| Photo 19. Import data to the 'System Identification Tool' | 49 |
| Photo 20. 'System Identification Tool' poles and zeros..... | 49 |

5. Notation

| | |
|------------------------------|---|
| A | Ampere |
| AC | Alternating Current |
| BLDC | Brushless Direct Current Electric Motor |
| DAC | Digital-to-Analog Converter |
| DC | Direct Current |
| EEPROM | Electrically-Erasable Programmable Read-Only Memory |
| ESC | Electronic Speed Control |
| Hz | Hertz |
| I/O | Input / Output |
| IR-LED | Infrared – Light-Emitting Diode |
| KB | Kilo-Byte |
| $\text{m}\cdot\text{s}^{-2}$ | meters per second squared |
| mA | Milli-Amperes |
| MHz | Mega-Hertz |
| ms | Milli-seconds |
| PCB | Printed Circuit Board |
| PI Controller | Proportional-Integral Contoller |
| PID | Proportional-Integral-Derivative |
| PSD | Position Sensitive Detector |
| PWM | Pulse-Width Modulation |
| SRAM | Shadow Random Access Memory |
| USB | Universal Serial Bus |
| V | Volts |

6. State of art and object

The design proposed in this thesis is commonly used in the academic world. There are many universities and projects with related topics, to a greater or lesser extent, with the system developed in this project. Physics, programming and control theory are required to achieve a good performance of the system, which gives this project a big academic interest.

Key factors on the used prototype are their hardware components: propeller, motor, electronic speed controller (ESC) and sensor. They will determine the behaviour of the system, and they will influence the control strategy to be applied.

Brushless motors are the most popular in aeromodelling world and this was the key inspiring the project, giving a better performance in comparison to brushed DC motors. The main reasons to use brushless motors are given by their performance and advantages compared to a brushed motor, being their use generalized in the aeromodelling world with a great variety of motors in the market. In general, both motors are giving a correct performance with a control signal and the usage of any of these in the project would have been valid.

To choose a controller, there are two main options: one is to use a specially designed controller for aeromodelling, for example a specific model for a quadcopter, or to use a general controller and interface it with a microcontroller/microprocessor, which is a better option for customized projects. The sensors can be accelerometers, gyroscopes, distance sensors, rotary potentiometers, among other depending on the application. For this thesis, a distance measuring sensor was chosen, against the option of choosing an accelerometer.

The main focus of this project is the design of a prototype of this system from scratch, where the movement will be proportional to the thrust given by the rotation of the propeller connected to a motor located at the end of the rectangular bar. Primarily the focus is in the design and selection of the components, as its construction and integration for the posterior control.

7. Introduction

In this bachelor thesis, the design and construction of a prototype motor-propeller-rectangular bar is described.

The unstable nature of the proposed system leads into an interesting challenge in order to define a proper control strategy to balance the system. Physics, programming and control theory are key points for the success of the project.

In this chapter, a global overview of the thesis is presented, indicating where to find each design step.

7.1. Content

The Project starts in the Chapter 8 defining the details concerning the components of the system, including the hardware and software. Their behaviour and most significant characteristics are explained to have an overall understanding of what is needed to create the system.

Chapter 9 describes the different types to control a system and different characteristics that have to be taken into account when designing a P, PI or PID controller. The main focus is on the PI control as this is the type of control implemented in this project.

Subsequently in Chapter 10, the behaviour of the system is analysed in order to obtain the plant of it. Several of the tests performed are explained together with some obstacles that had to be solved to continue with the project. In this chapter the main issues occurred when developing the PI control are illustrated.

Chapter 11 is showing the behaviour of the system after analysing the transfer function, PID Tuner and Simulink characteristics. Once the model is known, it is compared to the reference system and the results are explained.

Once the behaviour of the system is explained and all the information is obtained, in Chapter 12 the conclusions and possible future further steps are explained.

7.2. Gantt Diagram

To understand how the project was evolving, Diagram 1 is showing the different tests carried out from October until April:

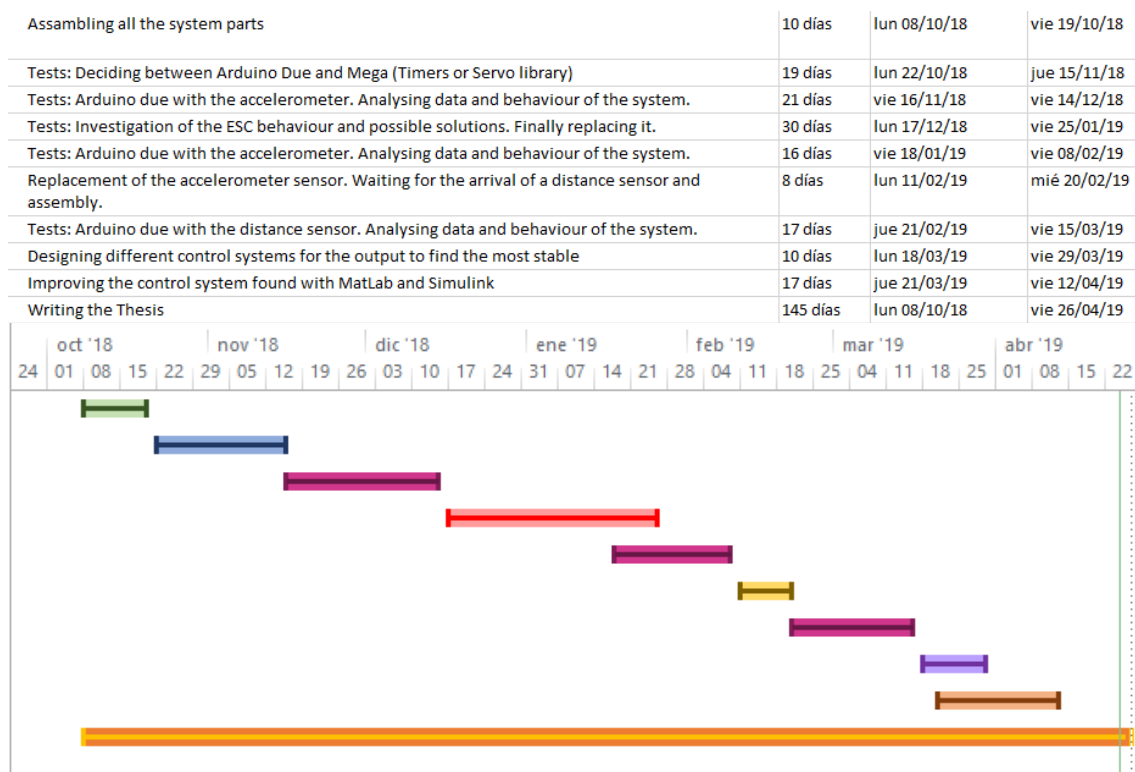


DIAGRAM 1. GANTT DIAGRAM

The first weeks were for assembling the different system parts and also to define the goals for the project.

After familiarizing with the system, there were several tests carried out to decide within the Arduino Due or Arduino Mega boards¹. The main tests were within using the Servo library² or Timers. At the end, due to some limitations for the servo library, timers were implemented in the code.

The next step was testing an accelerometer³ to control the system. Writing the code, performing the tests and analysing the output took about a month. While these tests were performed, some issues with the ESC⁴ were discovered. Because of this, the following month it was reconfigured but due to reasons explained in this project, it was finally replaced.

The accelerometer tests were taken again and the final decision was to also replace the accelerometer by another sensor, in this case it was for a distance measuring sensor⁵.

Finally the design of the controller was implemented and some more time was invested to improve the control system.

¹ See Chapter 8.7 for information about the Arduinos and Chapter 10.1 for the tests

² Servo library allows an Arduino board to control RC (hobby) servo motors

³ See Chapter 8.6 for information and Chapter 10.3 for the tests

⁴ See Chapter 8.3 for information and Chapter 10.2 for the tests

⁵ See Chapter 8.5 for information and Chapter 10.4 for the tests

8. Components of the system

This chapter is focused on the main components of the system motor-propeller-bar, including the structure, software and hardware. Beginning by describing in detail the plant and its behaviour for, at a later stage, describing the characteristics and behaviour of each component, as well as the specific model used for this project.

It has been taken into account that the election of the components is done to reuse some of the devices already available in the laboratory and also keeping in mind the possible future production of different projects based in this one. For that, it has been used different commercial components that can be easily accessible, economical and that can be acquired in the future without any difficulty.

8.1. Detailed information of the plant

Disposing of a system formed by a structure with an aluminium rectangular bar and a support joined together by a hinge which allows the system to rotate. Therefore, the part of the system with the rectangular aluminium bar is the mobile part and the support is the static part, allowing the rotation on the vertical axis. An image of the system can be seen in the Photo 1 whereas the static and mobile part are indicated on the Figure 1.

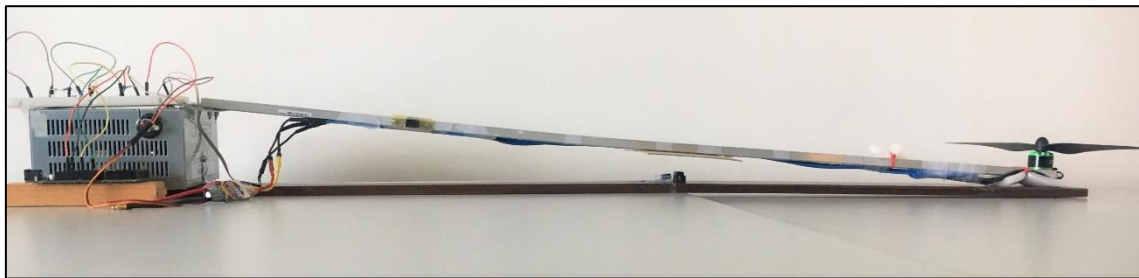


PHOTO 1. ENTIRE SYSTEM MOTOR-PROPELLER-BAR

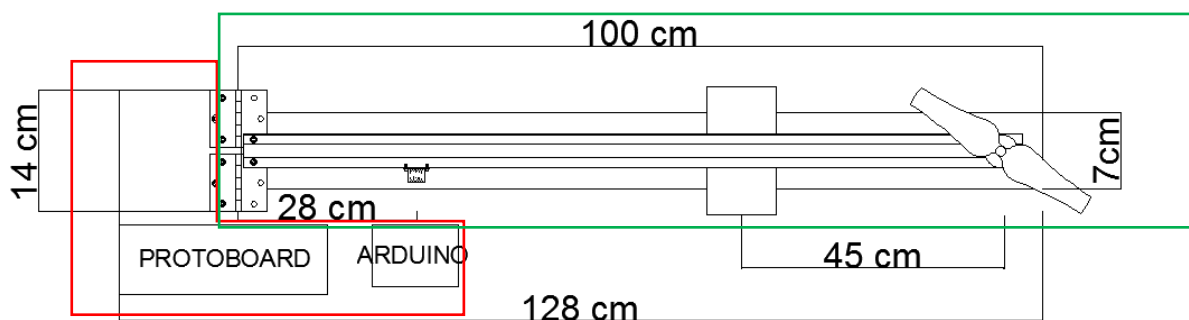


FIGURE 1. PLANT OF THE SYSTEM, IN RED THE STATIC PART AND IN GREEN THE MOBILE PART

The static part of this system is formed by a metal box and a wood bar support and a distance measuring sensor. The metal box, as it can be seen in the Photo 1, it is positioned on a wood rectangular bar. The mentioned wood support is limiting the movement of the moving part by being the horizontal limit where the aluminium bar is supported.

As shown in the Photo 3, 45 centimetres from the motor, on the support made of wood, there is located the distance measuring sensor, which is connected to the Arduino. Using the specifications, a condenser was added to avoid damaging the system. In order to read the measurement of distance correctly, the surface from where the sensor reads the data was improved by adding a thick layer of cardboard as shown on Photo 2.

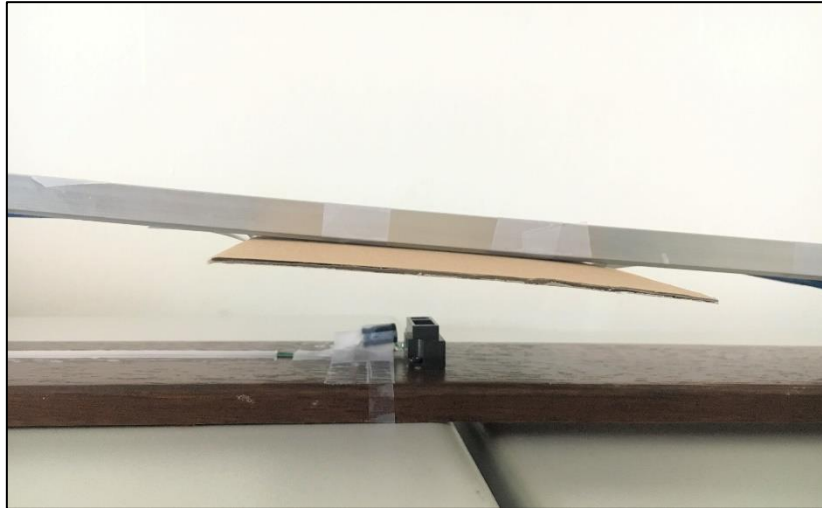


PHOTO 2. DISTANCE SENSOR WITH THE ADDED CARDBOARD

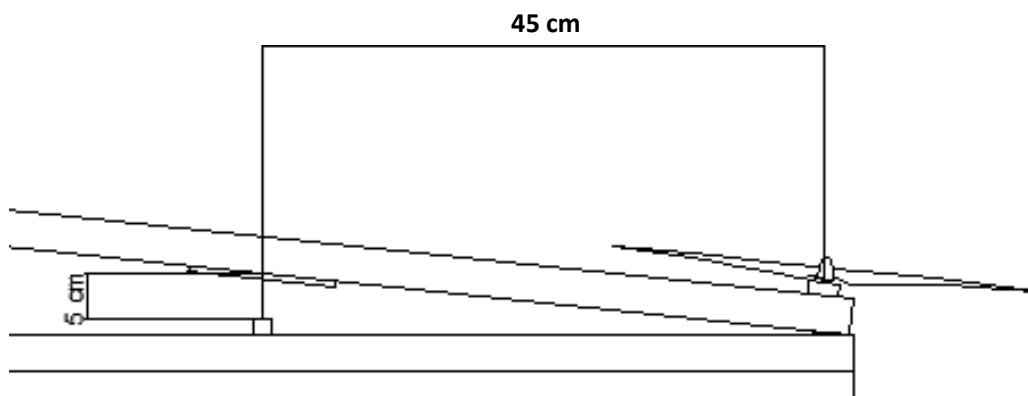


FIGURE 2. PLANT OF THE SYSTEM ZOOMING THE DISTANCE SENSOR

The moving group is formed by an aluminium bar with an accelerometer sensor attached to it, a motor and a propeller. The group motor-propeller is commercialized together and it is joined with three-phase cables. At the end of the aluminium bar the motor is fixed with screws. To avoid damaging the system when testing it, a thick layer of a padded material (*foam*) was positioned under the motor. On the opposite extreme of the aluminium bar, the hinge is allowing the rotational move on the vertical axis of the bar.

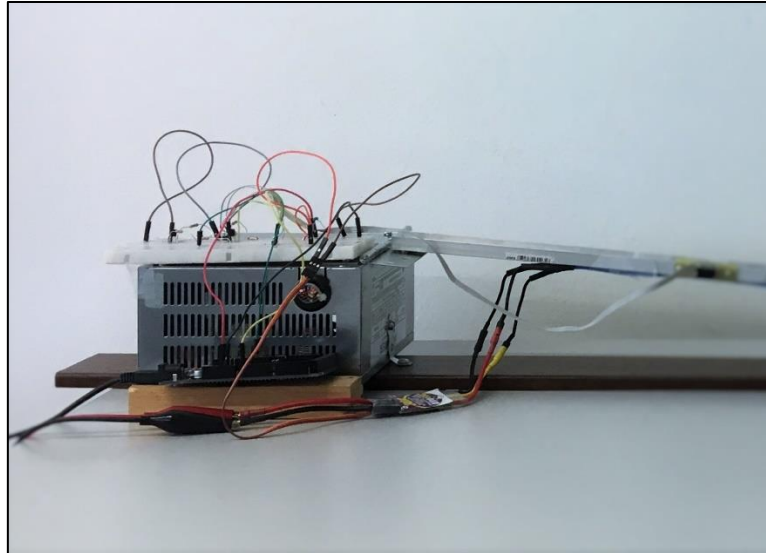


PHOTO 3. ROTATIONAL PART OF THE SYSTEM

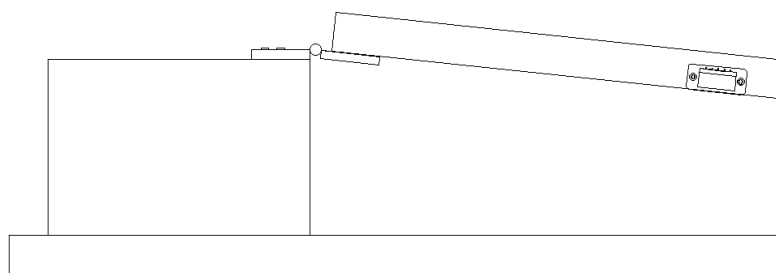
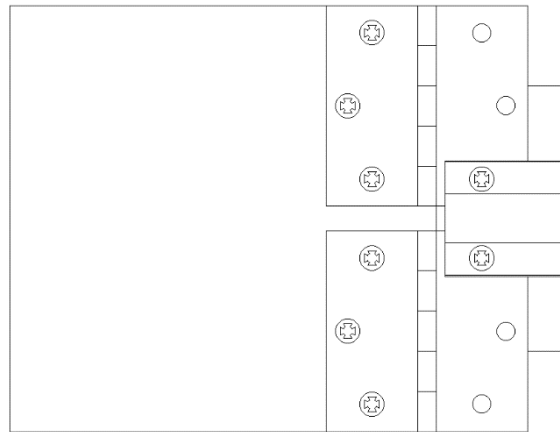


FIGURE 3. PLANT OF THE SYSTEM ZOOMING THE MOVING PART

Apart from the structural components, it is also necessary the usage of different electronic devices. One of the characteristics for controlling a brushless motor is that it requires an electronic circuit to operate. This electronic circuit is known as an ESC (Electronic Speed Controller) and it controls and regulates the speed of the motor using the PWM (Pulse Width Modulation).

The ESC is connected to the motor from one side by using three-phase cables and, from the other side, it is connected to the power supply and to the PWM signal. The PWM signal is an input signal generated, in this instance, from an Arduino device and it is modelling the signal to power the brushless motor at the desired speed. As it is a brushless motor, the power supply, as mentioned, is

not directly connected to the motor but connected to the ESC, and therefore, the ESC is in charge of powering the motor depending on the control signal given by the Arduino using a PWM pin.

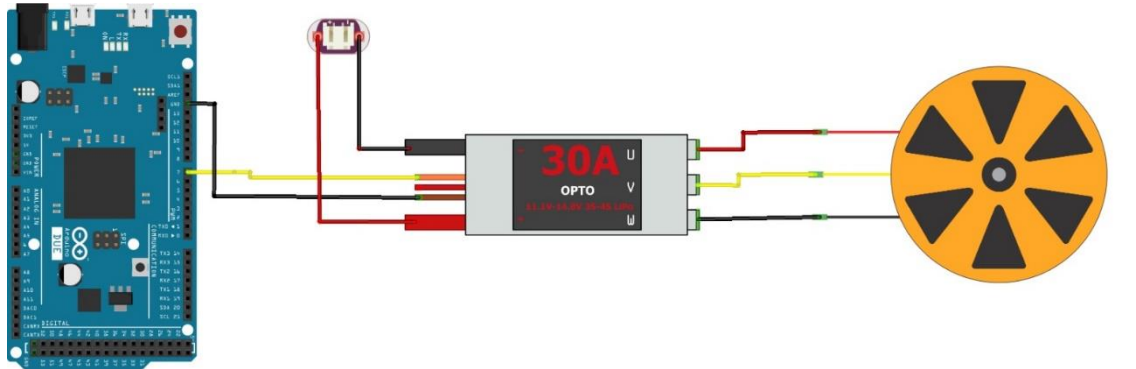


FIGURE 4. SCHEMATIC CONNECTION BETWEEN THE ESC-ARDUINO-MOTOR

The rotational speed control of the motor allows to control the thrust force exerted by the propeller, and with that, the position of the bar. If a particular position is ordered to reach, the propeller speed will have to vary depending on the position of the bar and the desired position that wants to be reached, this variation has to be determined by the system.

In this project the position is defined by the distance from the reference position, which is when the bar is held by the wood support. Therefore, the distance measuring sensor is determining the distance from the reference to the X-axis of the aluminium bar, implying a vertical movement of it.

This device is composed of an integrated combination of PSD (Position Sensitive Detector), an IR-LED (Infrared Emitting Diode) and a signal processing circuit. Even though the specifications are mentioning that the variety of the reflectivity of the object, the environmental temperature and the operating duration are not influencing easily the detection of the distance because it is adopting the triangulation method to have a greater robustness, a cardboard piece was located on the measuring surface of the sensor. This device outputs the voltage corresponding to the detection distance measuring from 4 to 30 cm, this is why it is located 45 cm from the propeller where the minimum distance measured is 5 cm.

To work with the data of the distance sensor we are using MatLab, but for interacting with the system it is needed another device, and for that, the chosen device is Arduino. The Arduino device is an electronic board that has its own microcontroller embedded with different input and output pins to connect to all kind of devices and sensors. The Arduino hardware is also incorporating an USB port type B to establish the connection between the computer and the board.

With all of these system and tools, we are able to receive the signal from the sensor and we can send the information with the PWM signal to power the propeller at the desired speed.

Further details on each component chosen, the principle of function and interactions among the system are explained in the following points.

8.2. Brushless Motor

The first component defined is the type of motor moving the propeller, depending on it, different components were chosen. The importance of knowing the behaviour of the motor and its characteristics is due to the influence it has over choosing the rest of components of the system.

8.2.1. Characteristics

Brushless DC motors are common in several applications across the world. Brushless DC motors, as its name is already mentioning, do not contain brushes and use direct current. It is not using brushes for changing the polarity, instead, it is using an external electronic circuit known as ESC (Electronic Speed Controller).

In this type of motors, the permanent magnets are located in the mobile part or rotor whereas the set of bobbins are located in the static part or stator, the opposite occurs in the brushed DC motors (Figure 5). The fact of using this brushless type is convenient because it is avoiding the friction between the rotor and the static part of the motor.

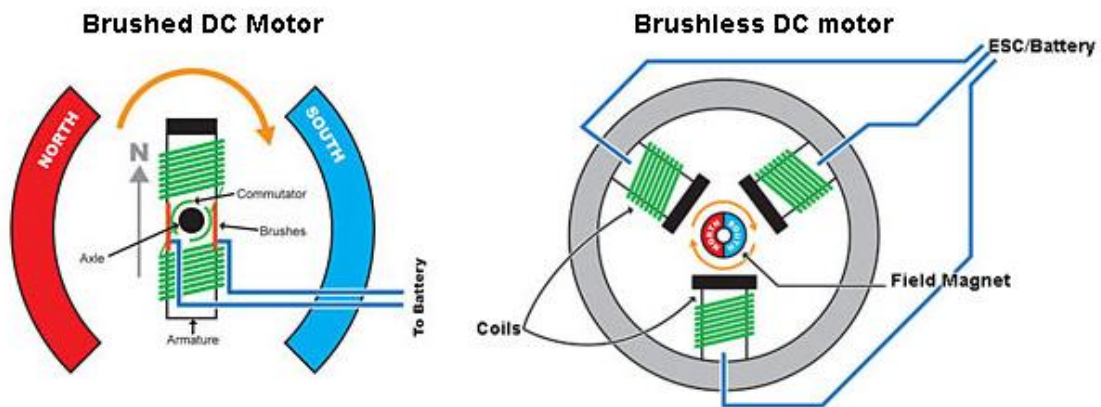


FIGURE 5. BRUSHED AND BRUSHLESS MOTOR SCHEMA (ANALOG.COM, 2019)

This is the way the brushless DC motors are increasing their efficiency by decreasing the heat loss. Moreover, their performance is consuming less power and allowing a higher range of speeds in the absence of mechanical limitations. The nonexistence of brushes is also decreasing the maintenance of the motor but on the opposite site, the main disadvantages are the cost of production, as well as the necessity of an ESC to control and operate the motor. This complex and expensive circuit is rising the price of the ensemble and, as a consequence, the final cost.

Being such a common motor in the radio control and aeromodelling world, there are a great variety of brushless motors and ESCs in the market, with different characteristics and prices. Therefore, it is an easy-to-find motor to fulfil the requirements of the project.

8.2.2. Operating the system

To understand the behaviour of the brushless motor it is needed to understand its structure and internal components.

There are two differentiable parts principally: the turning mobile part known as rotor and consisting of two permanent magnets, and the fixed part know as stator, consisting of two sets of conductive bobbins where the electricity is conducted.

Depending on the relative position between the rotor and stator, there can be two types of brushless motors: outrunner and inrunner. The outrunner type of motor spins its outer shell around its windings, meaning that the rotor is located in the outer part of the motor which is carrying the

magnets and in the inner part the set of bobbins is fixed. For the inrunner case, the stator is located on the outer part and in the inner part the magnets are found.

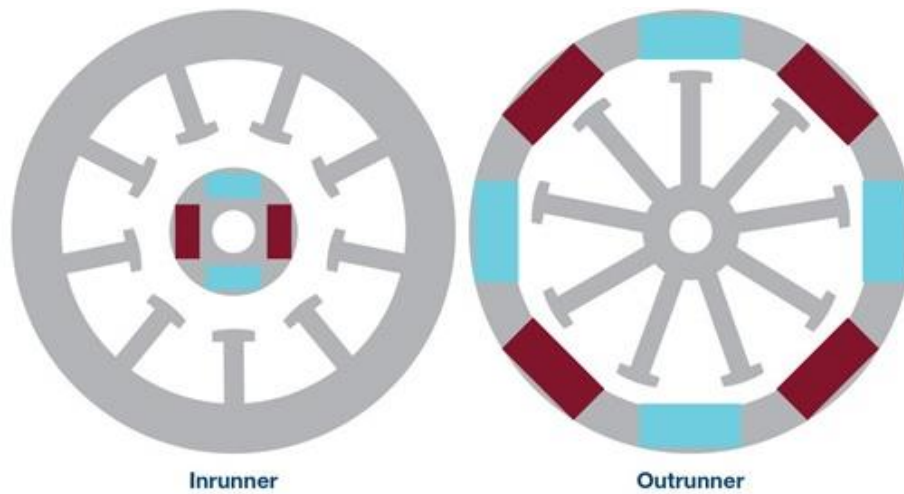


FIGURE 6. SCHEMA OF THE OUTRUNNER AND INRUNNER MOTORS (ANALOG.COM, 2019)

The working of the brushless motor is based on the electrical current that is defined as a permanent synchronous machine with rotor position feedback. The brushless motors are generally controlled using a three-phase power semiconductor bridge. The motor requires a rotor position sensor for starting and providing a proper commutation sequence to turn on the power devices in the inverter bridge. Based on the rotor position, the power devices are commutated sequentially every 60 or 120 degrees. For commutating the armature current it is using electronic commutation and for this it is an electronic motor, eliminating the problems associated with the brush and the commutator arrangement. The speed of the sequence is called frequency [Hz].

8.2.3. Chosen Motor: 2206-2150KV Multistar

The main requirement set at the beginning of the project was the:

- Usage of a brushless motor already available at the university
- Maximum power needed of 5V (due to the Arduino maximum power output)

This motor is used in other projects at the university and it is commonly known among students. Moreover, because of the good performance, its benefits and its generalised use, the chosen type of brushless motor is the outrunner. The motor is the 2206-2150KV Multistar and it is shown in the Photo 4. Even all its benefits, it's a discontinued motor and in case of replacement many points would have to be evaluated. ¡Error! No se encuentra el origen de la referencia. This model is already coming with the propellers and the datasheet can be found in the Annex VI.



PHOTO 4. BRUSHLESS MOTOR (HOBBYKING.COM, 2019)

8.3. Electronic Speed Controller or ESC

As mentioned in the previous chapters, the brushless motors require an electronic control to provide the power signal. This electronic controlling part is called Electronic Speed Controller or ESC, which is transforming the direct current (DC) to alternating current (AC).

An electronic circuit is used to change the speed of an electric motor. These are frequently used on radio controlled models that are electrically powered, and also, most frequently used for brushless motors providing an electronically produced 3-phase electric power low voltage source of energy for the motor. This controller is responsible for giving to the motor the signal to adjust the RPM, and consequently, controlling the speed of the motor.

The ESC usually consists of seven different connections for the right performance of the motor:

- Three outputs cables connected directly to the motor which are giving the control signal (3-BLDC motor phases)
- One negative (-) and one positive (+) LIPO connections
- One cable receiving the PWM signal to control the speed of the motor
- One cable connected to ground

Earlier, the speed controllers were mainly used in remote control boats and cars which used a variable resistor with a wiper that was stimulated by a servo motor. Currently, speed controllers switch the power ON and OFF as it is going to be explained in the following paragraphs. Moreover, ESC can be brushed or brushless. The brushed ESC was the first one and nowadays is a cheap device to work with. On the other hand, the brushless ESC is the modern advancement and it carries more power, a higher performance and it also lasts longer.

Without going into details of how the PWM⁶ signal works, the ESC needs to receive a signal that produces the movement of the motor. This signal is characterised because its pulse duration can vary between 1 and 2 milliseconds with an amplitude from 0 to 5 V. It is controlling the duty cycle, also called power cycle, which implies to control the pulse-width modulation (PWM) directly related with the control of the motor's speed transmitting the needed power. When the duration of the pulse-width is lower than 1ms, the motor remains immobile because it corresponds to the minimum speed of the motor, as this duration increases, the speed starts turning until a maximum of 2ms. The PWM (Figure 7) signal is generated by an Arduino board.

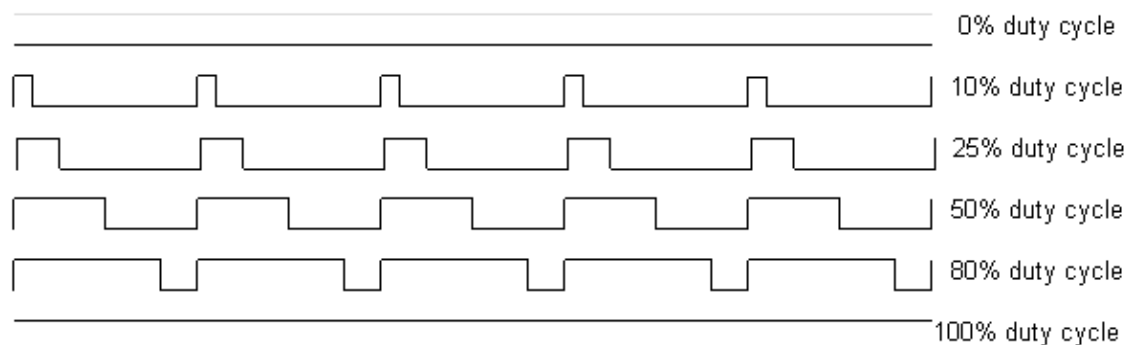


FIGURE 7. PWM SIGNAL FOR DIFFERENT PERCENTAGES OF THE DUTY CYCLE VALUES (ARDUINO.CC, 2019)

⁶ PWM refers to pulse-width modulation or pulse-duration modulation PDM, is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts.

As mentioned, the signal in the Figure 7 is received by the ESC and not the motor. The power signal received by the motor has different characteristics and will depend on the PWM signal and the power given to the ESC coming from the battery or power unit.

Ultimately, the ESC is an electronic device that is in charge of providing the necessary power to the motor taking into account the PWM received and the power supplied by the power unit or battery, to achieve different speeds to make the propellers turn.

8.3.1. Chosen ESC: Afro ESC 30A

The chosen ESC is due to the use of it in other university projects and it has been taken into account the consumption that requires the motor. This ESC can give up to 30A, even though we could have used a 5A or 6A ESC for this motor, the main reason to use this was because it was already in the laboratory. Its main disadvantages are that it is a discontinued model with a not up-to-date software. The model is called “Afro ESC 30 A” (Photo 5) and the datasheet can be found in the Annex V.

This model of ESC also comes with a USB Linker to update the software online, but as mentioned the website was not available; this USB has to be connected to the PWM input cable and ground when updating the software.



Photo 5. Afro ESC 30A (hobbyking.com, 2019)

8.4. Power Unit

To supply electrical power to the motor, it is needed a power unit or a battery with direct current that is going to be connected to the ESC. In this project the system is going to be powered by a power unit, working in this way is ensuring that the direct current is stable and constant during the whole test. Avoiding the problem generated by the batteries which can supply the power to the motor but can wind down when discharging.

Thus, the chosen method of powering the motor is a power unit of direct current, connected directly to the ESC through two banana cables, positive and negative. The supplier of the motor recommends to power the ESC from 9 up to 11V of DC.

8.4.1. Chosen Power Unit: Agilent 6573A DC Power Series

The chosen Power Unit is due to its availability in the laboratory, close to where the project was located and, because it is meeting the power requirements that were explained previously (11V DC). The model is the part of the 6570A series of the brand Agilent. The main characteristics are:

- Input: 230 Vac / 50 Hz
- Output: 0 – 30 V / 0-60 A
- Programmable

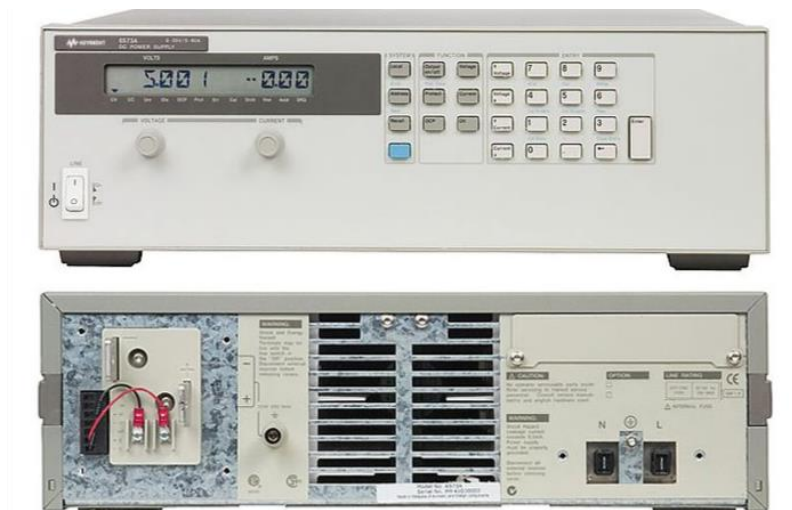


PHOTO 6. POWER UNIT USED TO PERFORM THE TESTS (KEYSIGHT.COM, 2019)

As it can be seen in the Photo 6, you can choose the program you want to use to work with the power unit or you can adjust it every time to your needs. In this case, the power unit has been programmed to give an output of 11 V to be able to work more efficiently and the intensity needed by the ESC is going to be given by the power unit.

8.5. Distance measuring sensor

A distance measuring sensor is an object with optical sensors using the triangulation measuring method. The most common infrared (IR) wavelength distance sensors which have an analogue voltage output are equipped with IR-LED (Infrared Emitting Diode) that can emit a narrow light beam. After reflecting from the object, the beam will be redirected to a second lens where a PSD (Position Sensitive Detector) can be found. The PSD has a different conductivity depending on the position where the beam falls. This conductivity is converted into a voltage that is digitalized and by using a DAC the distance can be calculated.

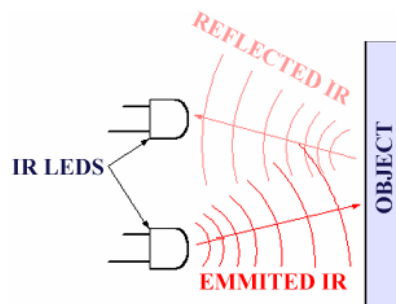


FIGURE 8. SENSOR WORKING PRINCIPLE


As it can be seen in the Figure 8, the moment IR light falls on the photodiode these output voltages and the resistances change in proportion to the magnitude of the IR light received.

All sensors have a specific measuring range where the measured results are creditable and this range depends on the type of the sensor. The maximum distance measured is restricted by:

- The amount of reflected light
- Inability of the PSD to register small changes of the reflected ray

If the object is too far, the output remains approximately the same. If the distance is too short, the peculiarity of the brand used in this project (*Sharp*) is the output starts to decrease steeply. This means that to one output value there are two values of distance, when the range of the sensor is known, this problem is easily avoided. In the Table 1, some sensors from the brand 'Sharp' with different operating ranges are shown.

TABLE 1. SHARP SENSORS WITH DIFFERENT OPERATING RANGES

| <i>Sharp sensor</i> | <i>Model</i> | <i>Range</i> |
|--|--------------|---------------|
|  | GP2Y0A21 | 10 to 80 cm |
| | GP2Y0A02 | 20 to 150 cm |
| | GP2Y0A710 | 100 to 550 cm |

8.5.1. Chosen distance measuring sensor: GP2Y0A41SK0F

As it has been explained in Chapter 8.5 the output of the sensor is inversely proportional, by meaning that when the distance is growing, the output is decreasing gradually. The graph of the relation between the distance and the output is in the datasheet of the sensor () and the voltage output is also presented in the Figure 9. The measuring range of the used sensor is from 4 to 30 cm.

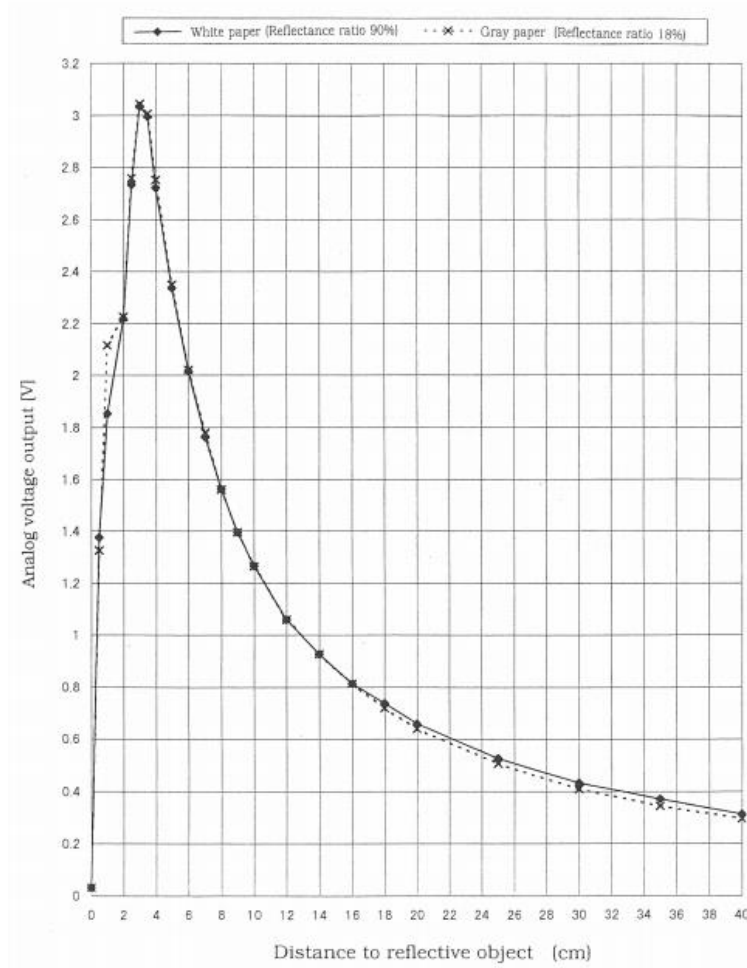


FIGURE 9. CHOSEN SENSOR OPERATING RANGE

In the experiment, the position is defined by the distance from the reference that is oriented horizontally. Therefore, the distance measuring sensor is needed to be determining the distance from the reference to the X-axis of the bar. Even though the specifications are mentioning that the variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because it is adopting the triangulation method, a cardboard was added to the reading surface. As this device outputs the voltage corresponding to the detection distance measuring from 4 to 30 cm, it is located 45 cm from the propeller.

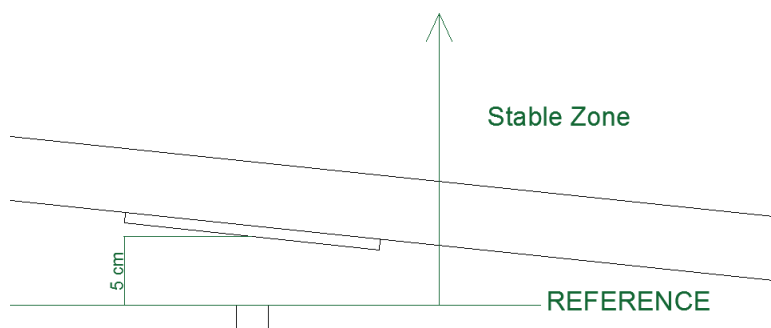


FIGURE 10. DISTANCE MEASURING SENSOR DETECTING AN OBJECT RANGE

8.6. Accelerometer

An accelerometer is a device to measure acceleration, which is the rate of change of the velocity of an object. The measure is in G-forces (g) or meters per second squared (m/s^2). A single G-force on planet Earth is approximately to 9.81 m/s^2 , varying slightly depending on the elevation.

Accelerometers are devices that can sense either static or dynamic forces of acceleration. Static forces refers to gravity while dynamic forces can include vibrations and movement, therefore, they are useful for sensing vibrations in systems or for orientation applications.

Effectively, an accelerometer contains capacitive plates internally, some of them fixed and others attached to minuscule springs that move internally as acceleration forces act upon the sensor. These plates move internally between each other which means the capacitance between them changes and the acceleration can be calculated from these changes. Other type of accelerometers can work with piezoelectric materials, these crystal structures output electrical charge when placed under mechanical stress like acceleration. Moreover, an accelerometer can measure the acceleration in three different axes, these are shown in the Figure 11.

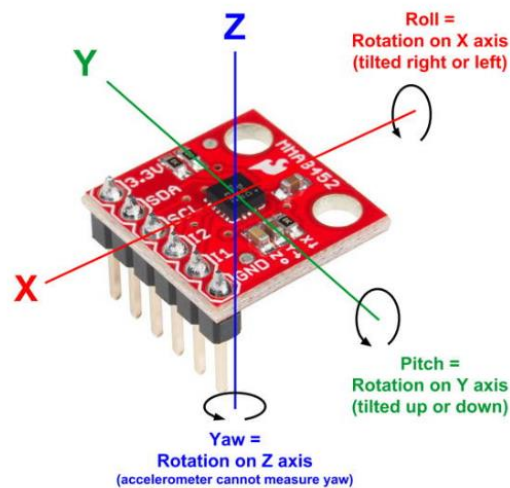


FIGURE 11. ORIENTATION OF THE 3 AXES OF AN ACCELEROMETER

8.6.1. Used accelerometer: MMA7341LC

The first idea in this project was to use an accelerometer to know the angle of the bar. The chosen accelerometer was the "MMA7341LC" from the brand RoHS/Freescale due to the use of it in previous university projects.

This accelerometer is a low power, low profile capacitive micro-machined accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature compensation, self-test, and g-Select which allows the selection between two sensitivities.

The accelerometer was located 23 centimetres from the hinge (Figure 12, 77 cm from the motor), there was a support to hold the accelerometer sensor as seen in the Photo 7. This was fixed by a plate with two screws joining this support to the main aluminium bar. The plate had the accelerometer sensor glued with mounting adhesive to the surface and using a thin paper to have the maximum adhesion possible.

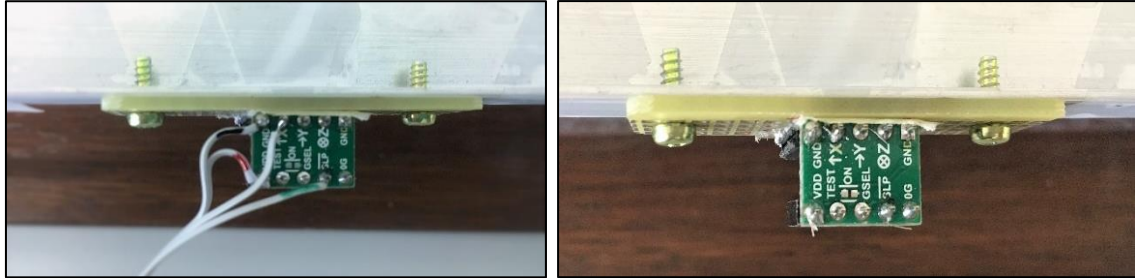


PHOTO 7. ALUMINIUM BAR WITH THE ACCELEROMETER (WITH THE WIRES CONNECTED ON THE LEFT, WITHOUT WIRES ON THE RIGHT)

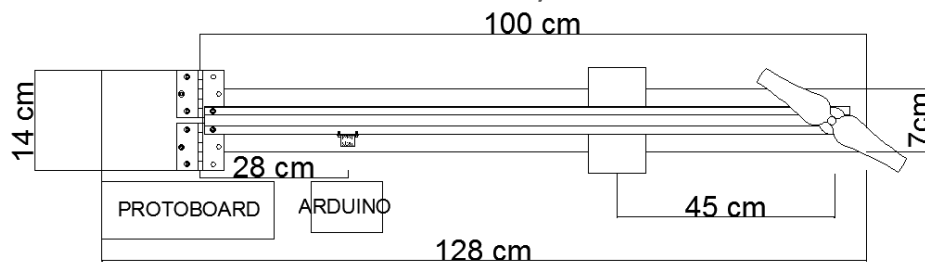


FIGURE 12. PLANT OF THE SYSTEM INDICATING THE DISTANCE FROM THE ACCELEROMETER TO THE HINGE AND MOTOR

Following the same principles as explained in the Chapter 8.1, the rotational speed control of the motor allows to control the thrust force exerted by the helix, and with that, the position of the bar.

The accelerometer worked similarly as the distance measuring sensor. In this case the position is defined by the angle regarding the reference. Therefore, the accelerometer sensor was needed to be determining the measurement of this X-axis. To analyse the data of the accelerometer it was also used MatLab, but Arduino was used to interact with the whole system. An image of the chosen sensor can be found in the Photo 8.

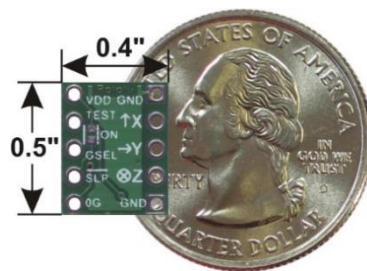


PHOTO 8. CHOSEN ACCELEROMETER

8.7. Arduino

Arduino is the name of an electronic platform created at the Ivrea Interaction Design Institute (Italy) which aims to be as easy-to-use as possible. The hardware and software are completely open-source; it is mostly aimed towards aspiring students without a background in electronics or programming. (Arduino.cc, 2019).

The Arduino is a board supplied by a microcontroller with different inputs and outputs and its own programming language. It can be connected to a computer and programmed in an easy way. Arduino boards are able to read inputs and turn them into outputs by using the microcontroller on it.

The software is free and can be found on the official Arduino website, it is the simplest way to develop a project involving a microcontroller, the board is already mounted, the bootloader ready to connect via USB to the computer and there are many libraries available online.

Arduinos come in many different varieties, for example the Arduino Uno that appears on the Photo 9. As the community has been changing, Arduino boards have adapted to the new needs and challenges. Most of them are based around the AVRseries of chips, which are manufactured by Atmel. These chips are providing the real functionality of the board, Arduino helps to use the board much simpler and easier. For this project, the real value of Arduino comes for the entire open-source ecosystem that has grown around the boards, the libraries and the easy communication between sensors and electronic devices.

Over the years, Arduino has been chosen to develop thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers has gathered this open-source platform and their contributions can be a great help when developing any kind of project. Arduino was chosen for this project because of its simplicity, easy to develop, it runs on Mac, Windows and Linux, it is a low cost device, interactive and open-sourced. In addition, there were already several and different Arduino boards available at the university.



PHOTO 9. ARDUINO UNO

8.7.1. Arduino Due

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU shown in the Photo 10. It is the first Arduino board based on a 32-bit ARM core microcontroller. Unlike most Arduino boards, the Arduino Due board runs at 3.3V. The maximum voltage that the I/O pins can tolerate is 3.3V. The board contains everything needed to support the microcontroller. It has to be connected using a micro-USB cable or power it with an AC-to-DC adapter or battery to get started. Arduino Due is compatible with all the Arduino shields⁷ that can work at 3.3V.

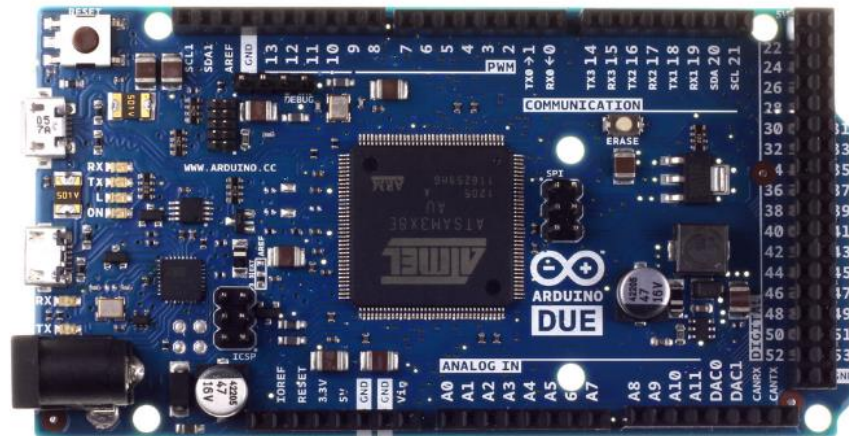


PHOTO 10. ARDUINO DUE

8.7.1.1. Arduino Due: Parts

In the Figure 13 there can be seen the different pins, connections and parts that are composing an Arduino DUE.

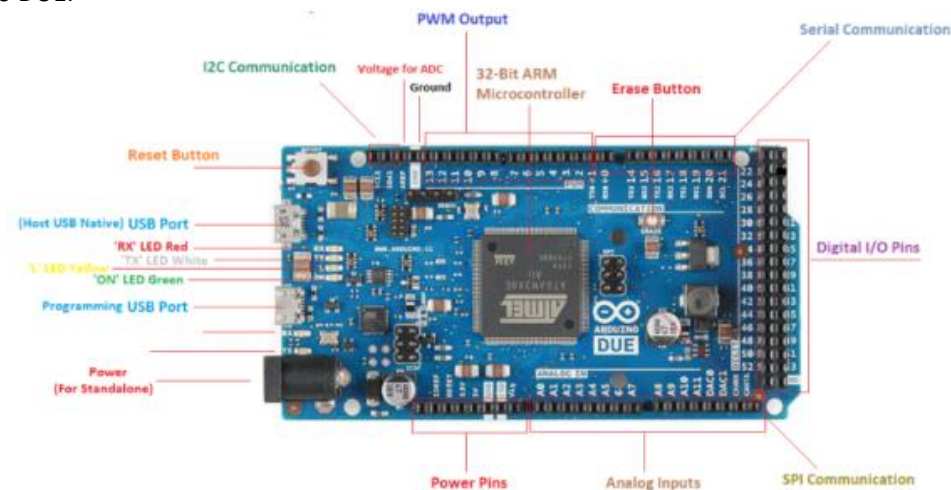


FIGURE 13. PARTS OF AN ARDUINO DUE

⁷ Arduino Shields are boards that can be plugged on the top of the Arduino PCB extending its capabilities

8.7.2. Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 shown in the Photo 11. The board contains everything needed to support the microcontroller. It has to be connected using a micro-USB cable or power it with an AC-to-DC adapter or battery to get started. The Arduino Mega is compatible with most of the shields designed for the Uno and the former boards Duemilanove or Diecimila.

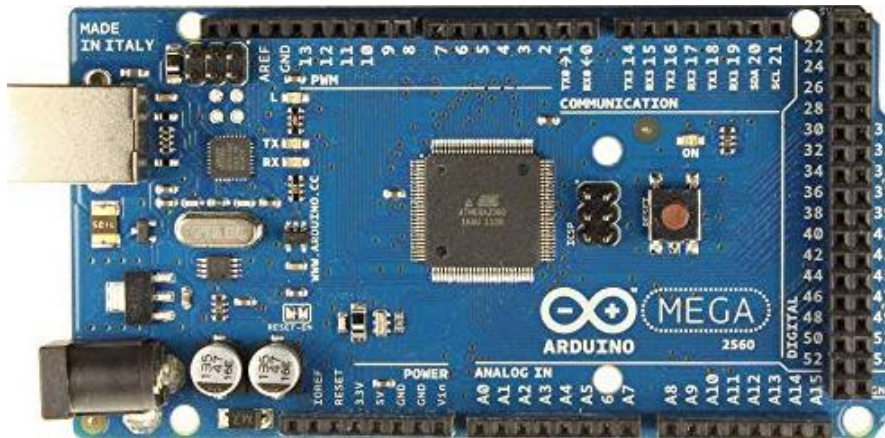


PHOTO 11. ARDUINO MEGA 2560

8.7.2.1. Arduino Mega 2560: Parts

In the Figure 14 here can be seen the different pins, connections and parts that are composing an Arduino DUE.

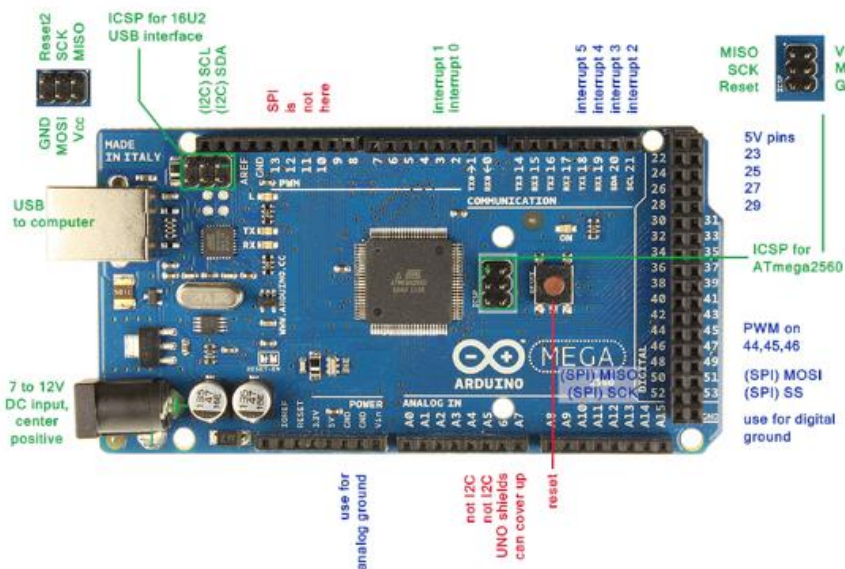


FIGURE 14. PARTS OF AN ARDUINO MEGA 2560

8.7.3. Comparison between Arduino Due and Arduino Mega 2560

In the following Table 2 there are shown the most relevant differences between the Arduino Due and Arduino Mega.

TABLE 2. COMPARISON BETWEEN ARDUINO DUE AND ARDUINO MEGA

| <i>Characteristics</i> | <i>Arduino Due</i> | <i>Arduino Mega 2560</i> |
|------------------------------------|--|---|
| <i>Microcontroller</i> | AT91SAM3X8E | ATmega1280 |
| <i>Operating Voltage</i> | 3.3 [V] | 5 [V] |
| <i>Input Voltage (recommended)</i> | 7-12 [V] | 7-12 [V] |
| <i>Input Voltage (limits)</i> | 6-16 [V] | 6-20 [V] |
| <i>Digital I/O Pins</i> | 54 (12 providing PWM output) | 54 (15 providing PWM output) |
| <i>Analog Input Pins</i> | 12 | 16 |
| <i>Analog Output Pins</i> | 2 (DAC) | - |
| <i>Total DC Output Current</i> | 130 [mA] | - |
| <i>DC Current per I/O Pin</i> | - | 40 [mA] |
| <i>DC Current for 3.3 V Pin</i> | 800 [mA] | 50 [mA] |
| <i>DC Current for 5 V Pin</i> | 800 [mA] | - |
| <i>Flash Memory</i> | 512 [KB] available for the user applications | 128 [KB] of which 4 [KB] used by bootloader |
| <i>SRAM</i> | 96 [KB] (two banks: 64 [KB] and 32 [KB]) | 8 [KB] |
| <i>EEPROM</i> | - | 4 [KB] |
| <i>Clock Speed</i> | 84 [MHz] | 16 [MHz] |
| <i>Cost</i> | 35 € | 35 € |

The main differences shown on the Table 2 are the operating voltage, the I/O pins, the memory size and also the speed. In this project the chosen option is the Arduino Due. The problem this Arduino board is presenting is that most of the standard shields work at 5 [V], so this must be taken into account when interacting with it, but does not suppose any problem for this project as shields are not used. On the other hand, for most of the electronic parts it can work without problems.

8.7.4. Chosen Arduino: Arduino Due

There are many different Arduino boards each one with some mutual and also different characteristics. In this project, as mentioned, the one used is an Arduino Due board. It has 54 pins that can be configured as a digital input or output between 0 and 3.3 V, among those, 12 can be used as a PWM output. It also has analogue inputs, a DAC, it is supplied with a reset button and like other Arduinos it can be easily connected to the computer.

The main reasons why this Arduino was chosen are:

- Availability of PWM output pin to send the ESC the correct signal to control the motor
- It can receive analogue signals which allows to read the sensor's output
- The analogue read resolution can be increased to 12 bits, which is not possible with the rest of the Arduino boards

8.8. MatLab

MatLab is a programming platform designed specifically for engineers and scientists with each own programming language. It allows to visualize and process signals in an easy way, also, to implement control system, simulation or identification of dynamic systems. It is also very useful to analyse data, create models, applications and developing algorithms.

8.8.1. Version of MatLab: R2018a and R2013a

The version of MATLAB used in this project is the R2018a, R2013a is the one available in the laboratory and the R2018a on other computers at the university.

8.8.2. Simulink and PID Tuner

MatLab includes a package of tools called 'Simulink' which is a block diagram environment for Model-Based Design. It supports simulation, automatic code generation and continuous testing of embedded systems. It allows the creation of a model where different blocks can be introduced such as lineal system, continuous or discrete blocks among others by using a graphical interface with block diagrams making it an easy and intuitive software. It allows exporting to MatLab main interface the data simulated.

Another interesting tool that MatLab is proportioning is the PID tuning. This is the process of finding the values of proportional, integral and derivative gains of a PID controller to achieve the desired performance and meet the design requirements.

9. Control system

A control system manages, commands, directs or regulates the behaviour of a device or system by using control loops. The concern of controlling a part of an environment known as plant or system in order to produce a desired product for society. The prior knowledge of the plant to be controlled is often the main problem when designing the control system as it is a critical point for the effectiveness of the control.

There are different principles applied when designing a control system like electrical, mechanical, and mathematical among others. The complexity of the usage of different disciplines is the one that makes the control engineering a multi-faceted engineering domain.

Control systems can be categorized as open-loop or closed-loop feedback controls depending on their architecture and the control method applied to control the system. An open-loop system is designed to meet the requirements by using a reference signal and drives the actuators that directly control the process output. Any kind of output feedback is not present in this system. On Figure 15 the structure of this system is presented.

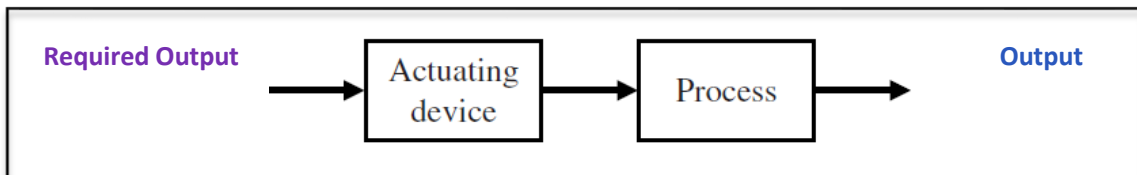


FIGURE 15. OPEN-LOOP SYSTEM GENERAL STRUCTURE

On the contrary, the closed-loop control systems take into account the actual output of the system and the desired output and feed it back to the controller to meet the requirement. This difference is commonly known as error signal and it is amplified and fed into the controller again as seen on Figure 16.

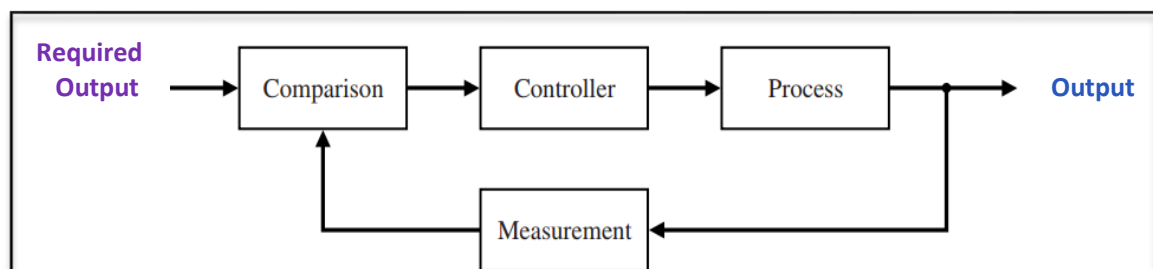


FIGURE 16. CLOSED-LOOP GENERAL STRUCTURE

9.1. P, PI and PID controllers

The controller stands for a comparative device that receives an input signal from a measured process variable, compares this value with the predetermined control point value (Set Point), and determines the appropriate amount of output signal required by the final control element to provide corrective action within the control loop. An electronic controller uses electrical signals and digital algorithms to perform its receptive, comparative and corrective functions.

9.1.1. P Controller

A P controller is the simplest algorithm in the PID family, P is for *proportional* or also called P-Only/P controller. Like all the automatic controllers, it repeats a measurement-computation-action procedure at every loop sample time (period, T) following the logic flow of other PIDs, like the one shown on the Figure 17.

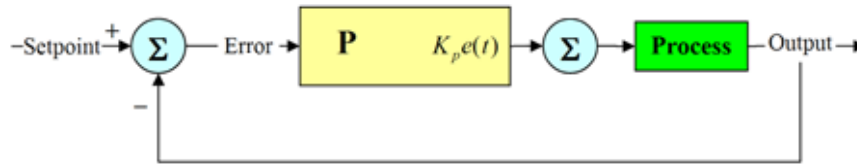


FIGURE 17. STRUCTURE OF A PROPORTIONAL CONTROL

The goal of the controller is to make the Error equal to zero ($e(t) = 0$), the error is equal to the Set Point minus the Actual Point Value (Sensor Measurement). Therefore, the proportional component depends only on the difference between the set point and the process variable, as shown in the following equation:

$$\text{Controller Output (t)} = K_p e(t) + p_0, \text{ where } p_0 \text{ is the controller output with zero error}$$

The main advantage of this control is it only has a parameter K_c , which is the proportional component, to be defined, this parameter defines how active or aggressive the controller will move when reading the error, and therefore, the proportional control can be recommended when needed for a fast-response system with a large transmission of coefficient. The disadvantage is it does not work for all the systems, the proportional control is not always sufficient.

9.1.2. PID Controller

A PID controller is a Proportional-Integral-Derivative (PID) control and it is the most common control algorithm used in the industry. Its popularity can be attributed to their robustness performance in a wide range of operating conditions and also their functional simplicity.

The basic idea behind this controller is to read the sensor and compute the desired actuator output by calculating three coefficients as presented in the Figure 18, these coefficients are the proportional K_p , the integral K_i , and the derivative K_d .

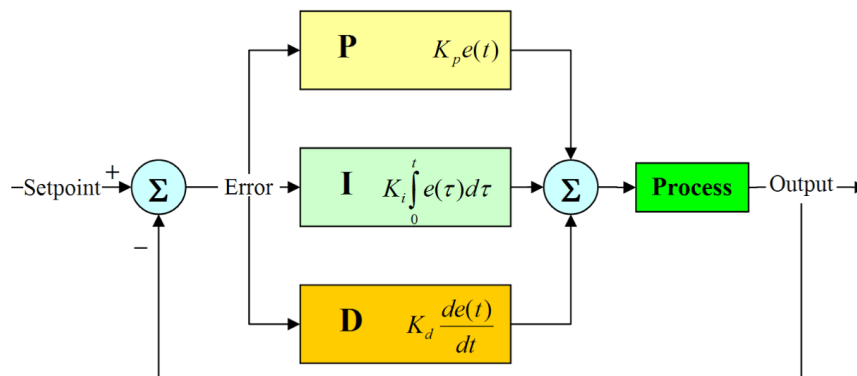


FIGURE 18. STRUCTURE OF A PROPORTIONAL-INTEGRAL-DERIVATIVE (PID) CONTROL

The theory shows evidence of the following: the higher the proportion coefficient, the less the output powers at the same control error; the higher the integration coefficient, the slower the accumulated integration coefficient; and the higher the differentiation or derivative coefficient, the greater the response of the system to the disturbance.

PID most common usage is in inertial systems with relatively low noise level of measuring channel. The PID has a fast warm up time, accurate set point temperature control and fast reaction to disturbances.

9.1.3. PI Controller

A variation of the PID control is to use only the proportional and integrals temps as PI control. A PI controller is a Proportional-Integral controller, the reason behind the extensive use of the PI controller is its effectiveness in the control of steady-state error of a control system. It has the disadvantage of the inability to improve the transient response of the system as it does not have the derivative part like the PID controller. The structure of the PI controller is shown in the Figure 19 and its equation is:

$$\text{Controller Output (t)} = K_p \epsilon(t) + K_i \int_0^t \epsilon(\tau) d\tau$$

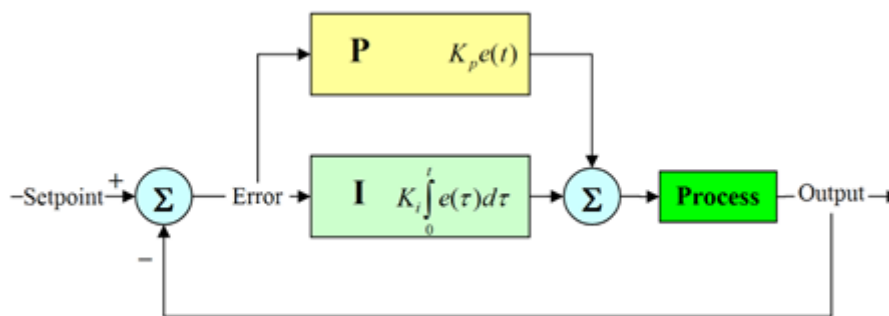


FIGURE 19. STRUCTURE OF A PROPORTIONAL-INTEGRAL (PI) CONTROL

Discrete PI controllers are implemented, as its name mentions, with a discrete sampling period and a discrete form of the PI equation. While the proportional term considers the current size $e(t)$ only at the time of the controller calculation, the integral term considers the history of the error, or how long and how far the measured process variable has been from the set point over the time.

9.1.4. Anti-windup

A common problem from the ideal PID implementations is the integral windup, due to the large change in the Set Point, the integral term can accumulate an error larger than the maximal value for the regulation variable (windup, thus, the system overshoots and continues to increase until this accumulated error is unwound. This is the reason why the anti-windup system is implemented. In the Figure 20 the schema of its structure for a PI control is shown:

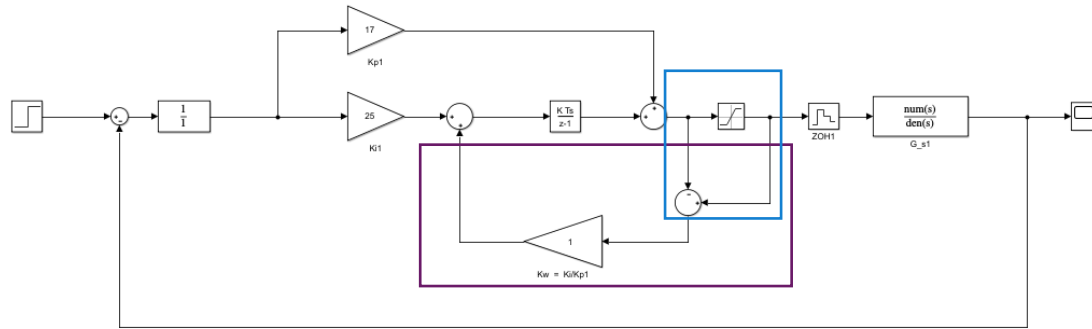


FIGURE 20. STRUCTURE FOR AN ANTI-WINDUP APPLIED TO A PI CONTROLLER (IN BLUE THE SATURATOR AND IN PURPLE THE ANTI-WINDUP)

The anti-windup is limiting the term that will be implemented in the next error. For this, a constant K_b or K_w is used, depending on the type of control implemented, the constant will have a different formula. In all cases it is calculated from the controller constants (K_p , K_i , K_d).

9.1.5. Saturator

The Saturator is way of delimiting the limits in the behaviour of the controller, the one shown in the previous subchapter (Figure 20) is referring to give limitations directly to the system by recalculating the error implemented in a loop. In some cases, this is not enough and another limitation are introduced, in this case the saturator is limiting the maximum and minimum power that can be applied to the circuit.

This means that when the power calculated exceeds the maximum or minimum values of the saturator, it will take those instead of the calculated ones. It is the simplest way to overshoot the problem that can present some ideal PID control systems. The structure is shown also in the previous Figure 20.

9.2. Digitalization of a controller

In this project, it is needed to digitalize the PI circuit from continuous to discrete as it has to be introduced in the Arduino. It is possible to do it by replacing the s complex variable for a z discrete equivalent expression.

There are several ways of finding this expression, in this work the Forward Euler is chosen, but it is also existing a Backwards Euler and Trapezoidal. This is done by following the transformation in the Chapter 10.5, it is also explained how it is done step by step.

10. Behaviour of the system

Before starting to design the PID control, the most important thing is to be able to interact and know the system to achieve a reliable plant and finally design the controller.

To analyse the sensor output, firstly it was tested which Arduino board would be more suitable to use, the tests carried out to decide the board utilized are in the Chapter 10.1. After that, the ESC-motor system had to be completely controlled by the user and the ESC behaviour had to be studied, these tests are found in the Chapter 10.2. Finally, the tests with the chosen sensor were performed and due to reasons explained on Chapter 10.3 and Chapter 10.4, the initial sensor was replaced.

Initially the system is expected to work by using the sensor's output to correct the angle of the bar. In the system, it can be found stable and unstable areas as shown in the Figure 21. The areas with the number '1' and '2' are the stable parts where the movement is controlled whereas the areas with the number '3', '4' and '5' are unstable.

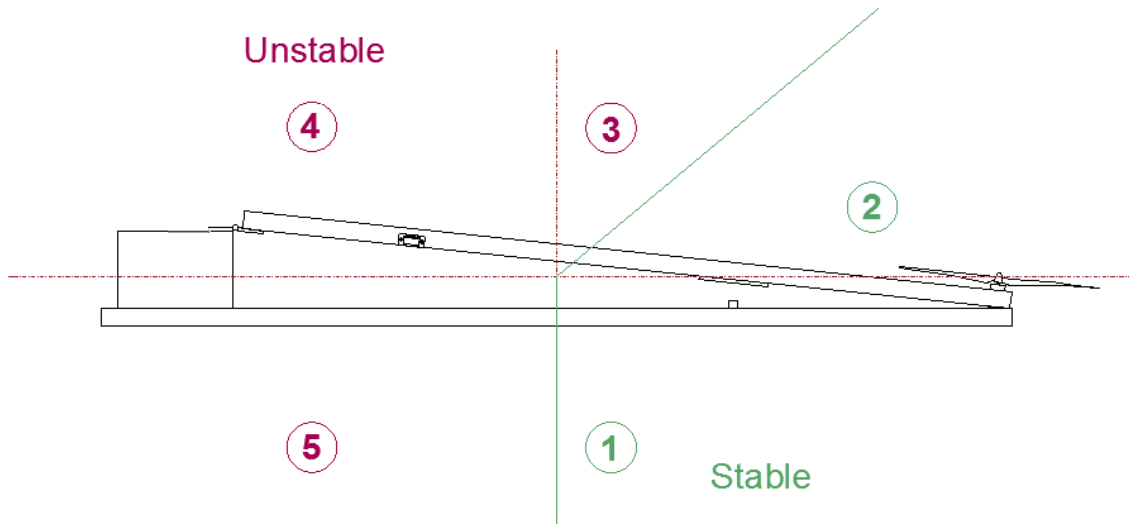


FIGURE 21. STABLE (GREEN AREAS, '1' AND '2') AND UNSTABLE ZONES (RED AREAS, '3', '4' AND '5')

10.1. Arduino

The first tests performed were as a result of the Arduino DUE board was not giving the voltage required by the ESC (5 V), therefore the Arduino MEGA board was also tested for this project. In the Chapters 10.1.1, 10.1.2 and 10.1.3 the performed tests are presented and the reasoning for the final decision.

10.1.1. Test 1: Arduino Due

As mentioned previously in the Chapter 8.7, Arduino DUE board can give an output of 3.3 V. The reason to start with a DUE board was arising from the usage of it in other university projects and because of its high analogue reading resolution.

The signal was generated as expected⁸ but the ESC was no able to work only with the Arduino DUE 3.3 V as expected and in order to keep the project simple, Arduino MEGA was taken into account.

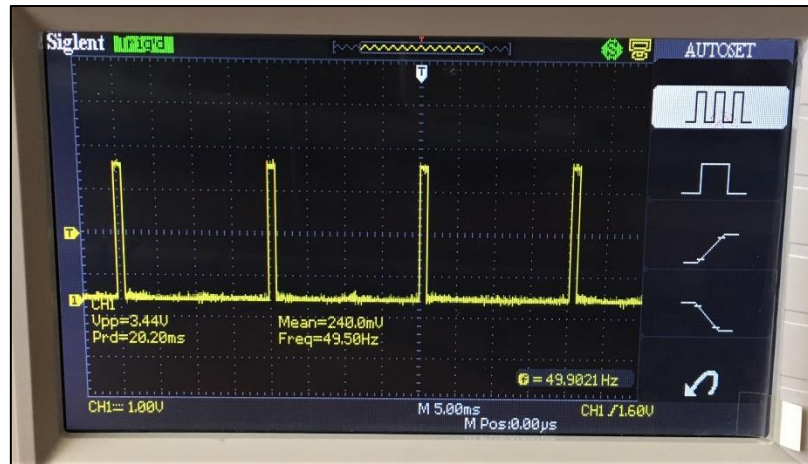


PHOTO 12. SIGNAL GENERATED BY THE ARDUINO DUE (3.3 V)

10.1.2. Test 2: Arduino Mega

During the second test, the problem of supplying the ESC with enough voltage was solved, but when reading the data and analysing it the resolution was not good enough (the range was deficient) to be able to design a PID controller. As there is no command for Arduino MEGA to increase the analogue reading resolution, the Arduino DUE was taken into account again.

10.1.3. Test 3: Arduino Due with an amplifying circuit

The Arduino DUE as mentioned could only give 3.3 V, hence there was an amplifier circuit implemented on the breadboard (see Figure 22). The problem regarding the data resolution was also avoided when using the DUE board as it already has a function to have a higher analogue reading resolution as it can be seen in the Annex III, the mentioned function is:

> `analogReadResolution(12);`⁹

⁸ See Chapter 10.1.3.1 and Annex III for more information about generating the signal

⁹ The Due, Zero and MKR Family boards have 12-bit ADC capabilities that can be accessed by changing the resolution to 12. This will return values from `analogRead()` between 0 and 4095. The resolution can be set higher than 12 but values returned will suffer approximation.

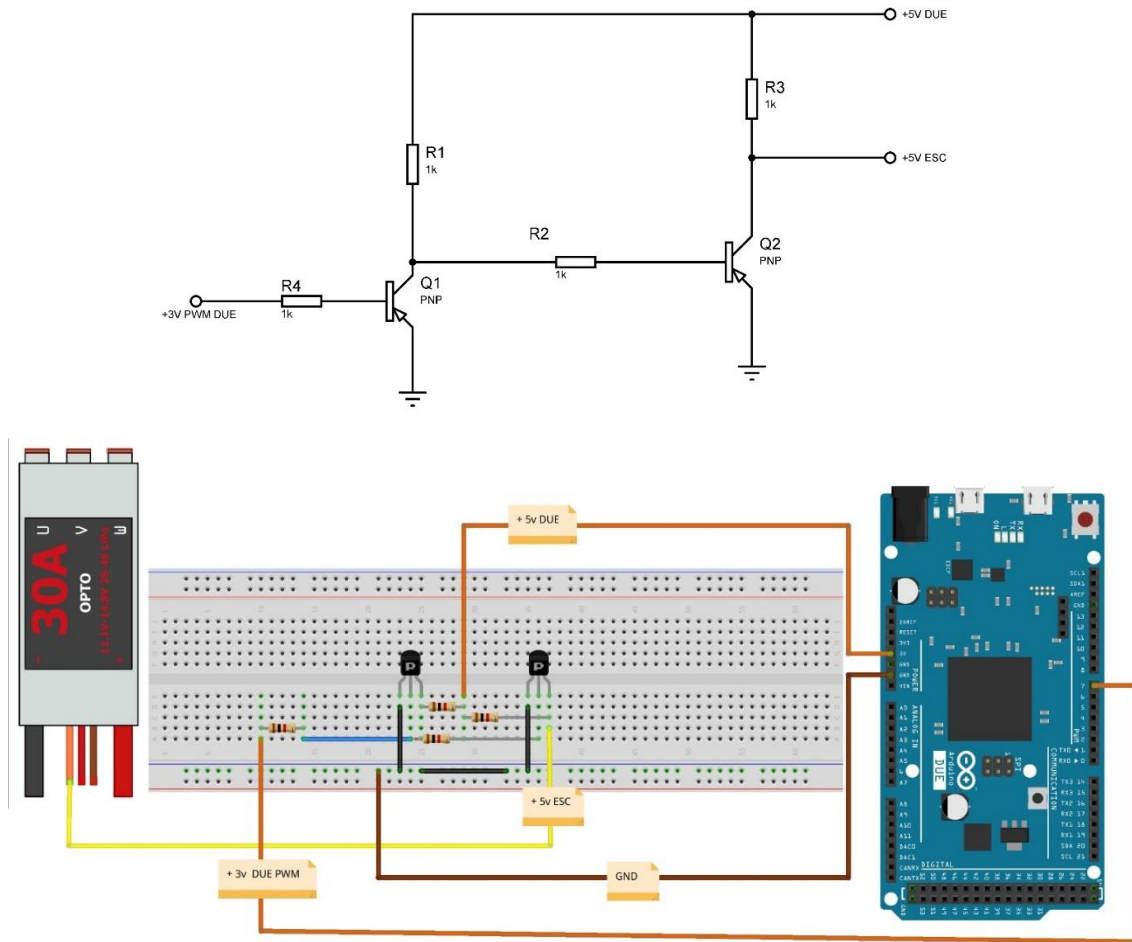


FIGURE 22. AMPLIFIER CIRCUIT (TOP) AND AMPLIFIER CIRCUIT CONNECTIONS (BELOW)

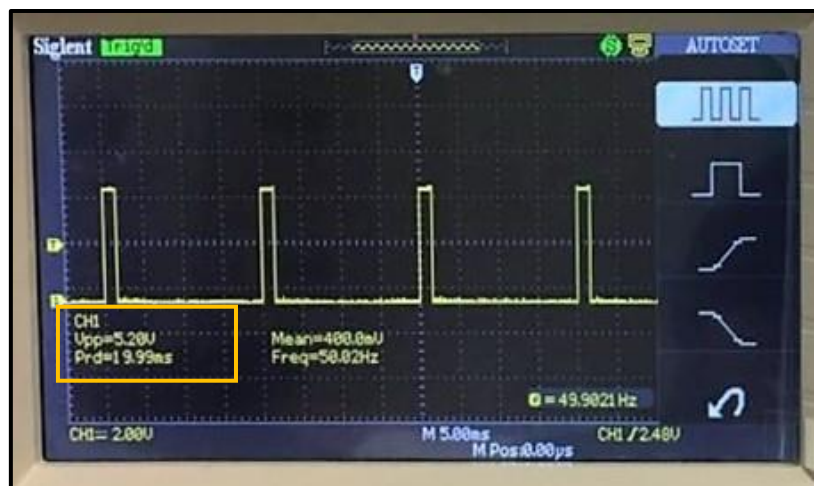


PHOTO 13. SIGNAL GENERATED BY THE ARDUINO DUE AND THE AMPLIFIER CIRCUIT (5 V)

10.1.3.1. Arduino Due: Generation of the signal

The following flowchart (Figure 23) is presenting the idea on how the generated signal is implemented and the code can be found in Annex III.

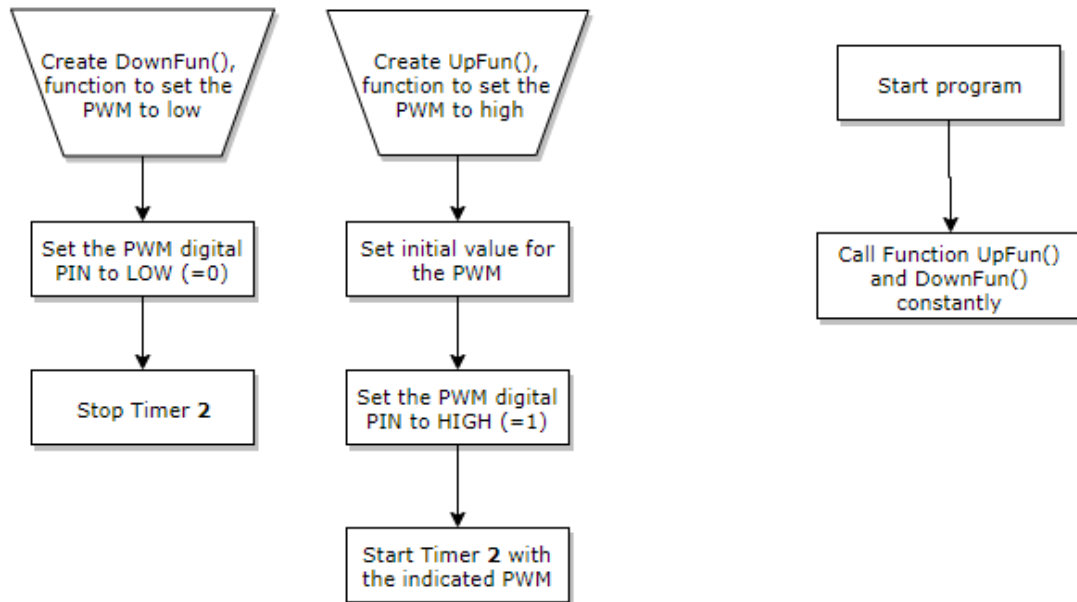


FIGURE 23. GENERATION OF THE SIGNAL FLOWCHART

The signal is generated by timers which are calling functions that are setting the PWM output to HIGH and LOW.

This signal generated:

- Period, $T = 20$ ms (Figure 24)
- $V_{pp} = 5$ V (aprox.)
- Frequency = 50 Hz

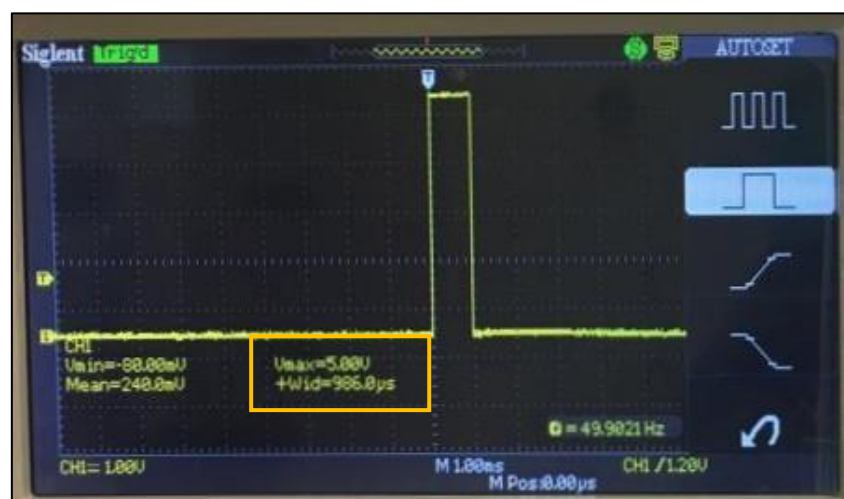


FIGURE 24. OSCILLOSCOPE - SIGNAL GENERATED BY THE ARDUINO

10.1.3.2. Arduino Due: Reading sensor's output

The following flowchart (Figure 25) is presenting the idea on how the reading the sensor's output is implemented and the code can be found on Annex III.

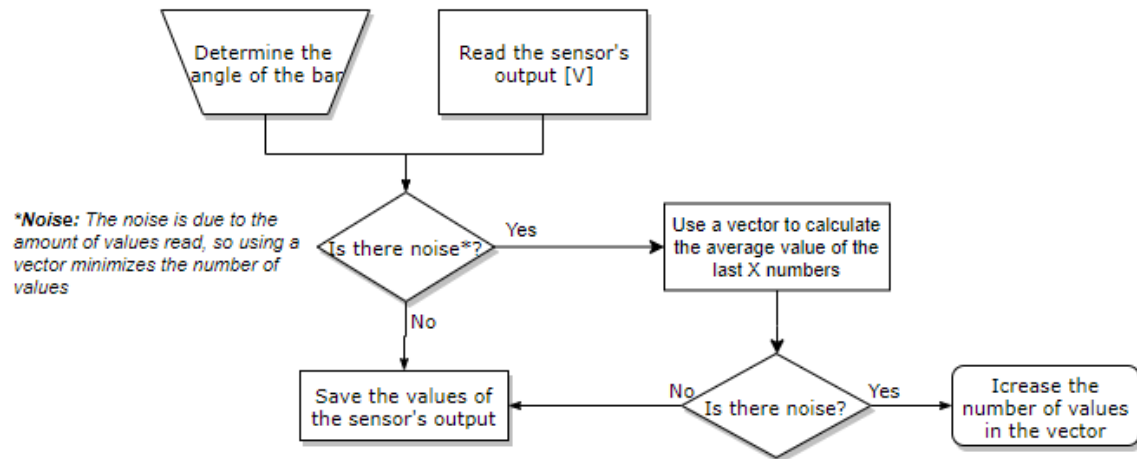


FIGURE 25. READING SENSOR'S OUTPUT FLOWCHART

As it can be seen in the Figure 25, the output is read every $1 \cdot 10^{-6}$ seconds.

10.2. Behaviour of the ESC

The ESC is an electronic device in charge of providing the necessary power to the motor. Taking into account the PWM received (in this case is coming from the Arduino Due board) and the power supplied by the power unit, it provides the power to the motor to achieve different speeds. The speed of the motor is making the propellers turn hence, elevating the bar. Therefore, the expected behaviour is to give the correct duty to the brushless motor.

It is important to take into account that the ESC is controlled by a pulse from 1 to 2 ms ($1060 \mu\text{s}$ to $1860 \mu\text{s}$).

10.2.1. Test 1: Working Range

After performing some tests with the ESC, the signal generated was correct as seen in the previous Figure 24. The problem was the working range of the motor, which was not a wide range. This influences in the amount of useful data that could be analysed afterwards.

It was only allowing '4' positions, by meaning that the PWM could only take four different values and the bar was already in the unstable zones (previous Figure 21).

10.2.2. Test 2: Reprogramming

Following the datasheet of the ESC¹⁰, it comes with a USB programming tool like the one shown in the Photo 14.



PHOTO 14. AFROESC USB PROGRAMMING TOOL

This USB combined with the SimonK software should be updating the controller and it can also be reprogrammed by the user. It is important to know the Afro ESC is from 2014 and it is not in stock in 2019, also, the SimonK software website was found under construction, as shown in the Photo 15:

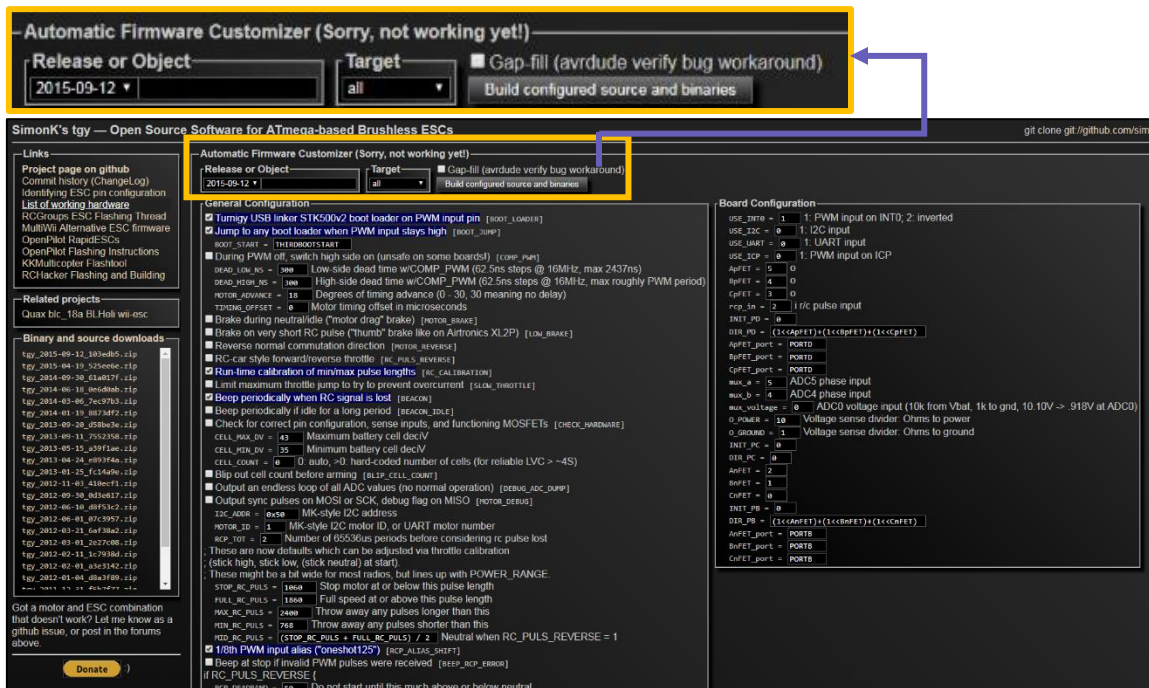


PHOTO 15. SIMONK ONLINE PROGRAMMING TOOL

SimonK is the most widely recognized ESC firmware in the *multicopter* world. Due to the unavailability of its usage during this project, another option to flash¹¹ the ESC has been supported. This one is called BLHeli, it supports many different ESC and allows to configure the ESC without a programming card. It just needs to connect the ESC via serial-to-USB interface.

A driver and the BLHeli program had to be downloaded. The driver can be easily downloaded from the internet website of 'Silicon Labs', the name is "CP210x USB to UART Bridge VCP Drivers" and it is

¹⁰ The AfroESC series is shipped with SimonK software, suitable for multi-rotor use without the need to program or adjust settings. (...) The easiest way of updating is with the Afro USB Linker from HobbyKing attached to the regular PWM input cable.

¹¹ "To flash" an ESC is considered to have the same meaning as "to update" the controller.

free to download. The BLHeli software can also be found easily online. Under the Figure 26 it is shown the interface of the program as it appears without any setup.

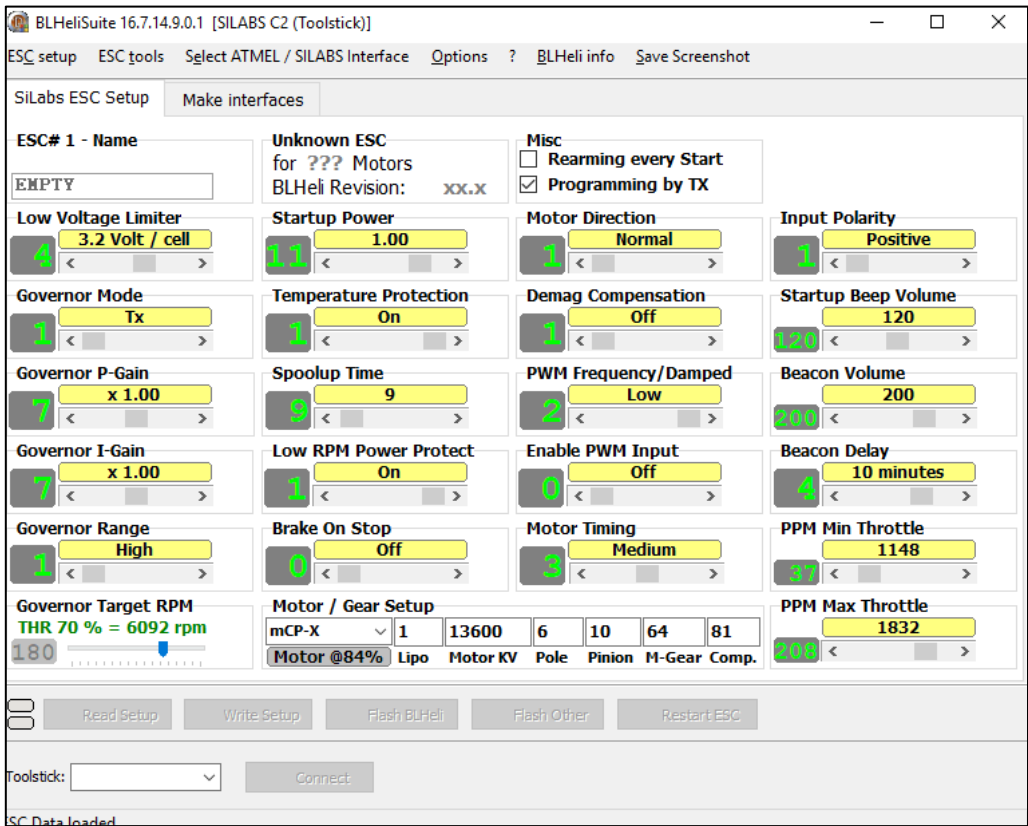


FIGURE 26. BLHELI INTERFACE

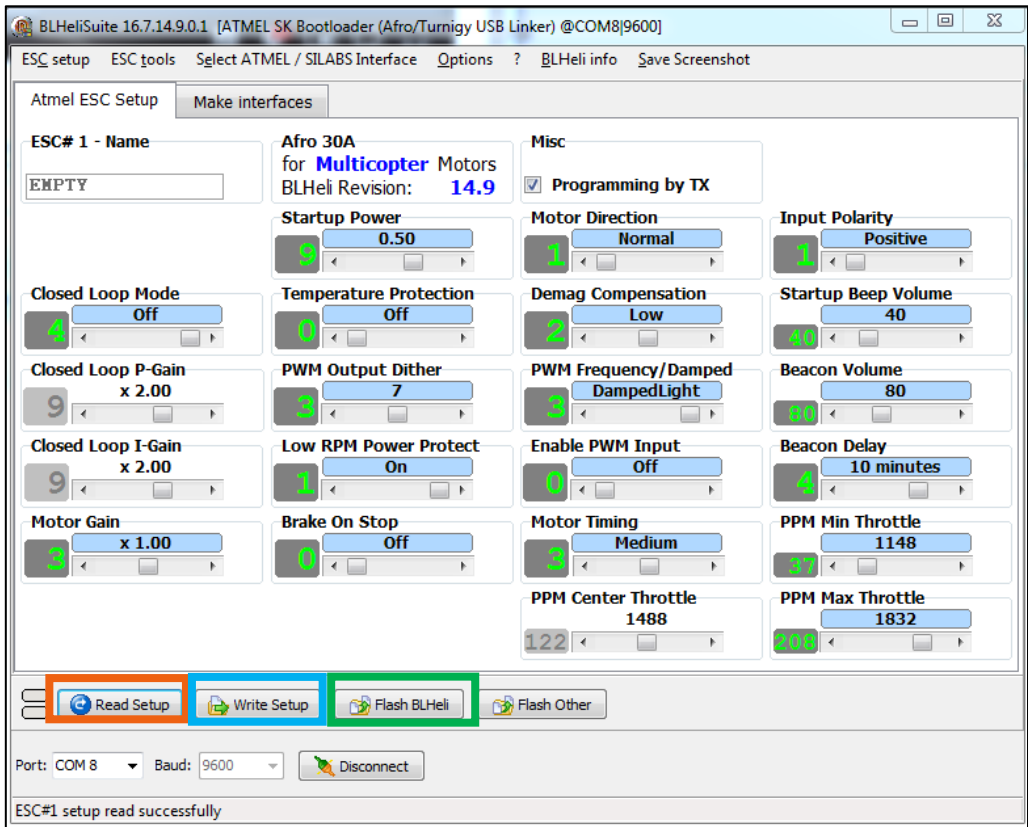


FIGURE 27. INTERFACE FOR THE AFRO ESC OF THE BLHELI SOFTWARE

To achieve the interface of the Figure 27, the first step is shown in the following Figure 28. The interface we need to interact with has to be selected as it appears in a square in orange. This interface has to match with the ESC used, and in this case it is the '5: ATMEL SK Bootloader (Afro/Turnigy USB Linker)'. The following step is to connect the 'Port', in this case it was 'COM 8' and afterwards press 'Read Setup' (Orange Square on Figure 27). This will show how the ESC is currently programmed.

The BLHeli software has a very interesting tool which is 'Flash BLHeli' (Green Square on Figure 27), this is showing a list of predetermined ESC features for different ESCs. In this list the 'ESC Afro 30A' option was chosen. The last step is 'Write Setup' (Blue Square on Figure 27), this will update the ESC as it appears on the program interface.

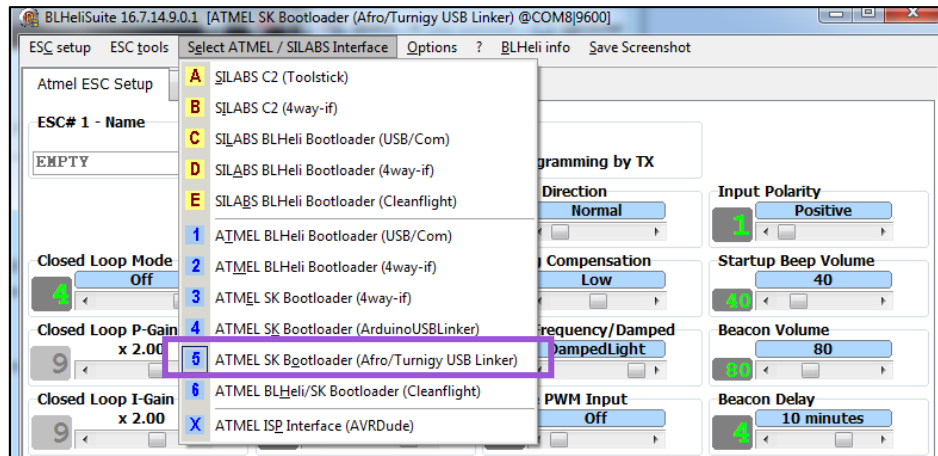


FIGURE 28. CHOOSING INTERFACE OF THE BLHELI SOFTWARE

After performing different tests with different configurations by playing with the 'PPM Min and Max Throttle' and the 'Start-up Power' options basically, there were some changes observed in the behaviour of the ESC, but the useful operating range was still too limited to implement a PID control.

The final solution was to replace the ESC by another one available at the university. This other ESC was the same brand, model and year (2014) and used in other projects. By this, the range problem was solved even though it was still small.

10.3. Accelerometer

The accelerometer was chosen in order to reuse the sensor from an old project as mentioned previously. To easily explain the tests carried out, Table 3 shows the pins of the accelerator and its functions. Figure 29 is showing the connection between the motor, Arduino Due and the accelerometer while Photo 16 is showing how the sensor was positioned.

TABLE 3. ACCELEROMETER SENSOR PINS AND FUNCTIONS

| Pin Name | Function | Name | Connection to the breadboard |
|----------|---|------------------|--------------------------------|
| 1 | No internal connection. Leave unconnected | N/C | - |
| 2 | X direction output voltage | X _{OUT} | Arduino (A0) – Analog Read Pin |
| 3 | Y direction output voltage | Y _{OUT} | - |

| | | | |
|----|---|--------------------|--------|
| 4 | Z direction output voltage | Z_{OUT} | - |
| 5 | Power Supply Ground | V_{SS} | Ground |
| 6 | Power Supply Input | V_{DD} | Power |
| 7 | Logic input pin to enable product or Sleep Mode | \overline{Sleep} | Power |
| 8 | No internal connection. Leave unconnected | N/C | - |
| 9 | No internal connection. Leave unconnected | N/C | - |
| 10 | Logic input pin to select g level | g-Select | Power |
| 11 | No internal connection. Leave unconnected | N/C | - |
| 12 | No internal connection. Leave unconnected | N/C | - |
| 13 | Input pin to initiate Self-Test | Self-Test | - |
| 14 | No internal connection. Leave unconnected | N/C | - |

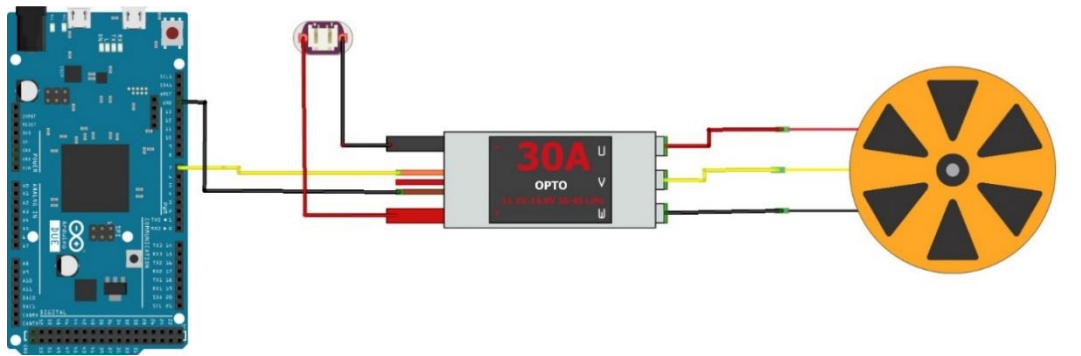


FIGURE 29. CIRCUIT IMPLEMENTED MOTOR-ACCELEROMETER-ARDUINO

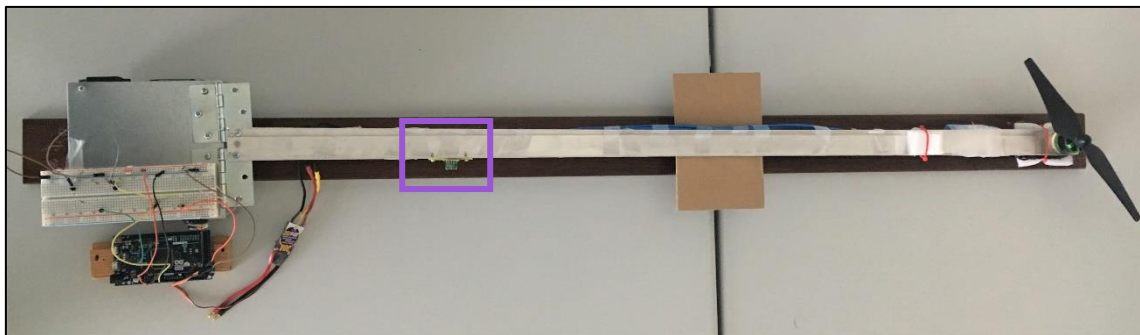


PHOTO 16. PLANT OF THE SYSTEM INDICATING THE ACCELEROMETER

10.3.1. Test 1: Performing the previous tests using the output of the accelerometer

Several tests were taken before analysing the data, as a result of the amount of tests performed between the Arduinos and ESC. Prior to analyse the data, the tests taken included adding different weighted pieces located in several positions to have a better control over the system but it was still

very unstable. Also, different vector sizes were considered to continue with the following step (Chapter 10.3.2).

10.3.2. Test 2: Analysing the output of the accelerometer

After the output data stored was satisfactory it was analysed using MatLab. This was done by plotting the data taking into account: Y-axis-left part with the output from the sensor, Y-axis-right part with the PWM and X-axis with the time, as shown in the Figure 30.

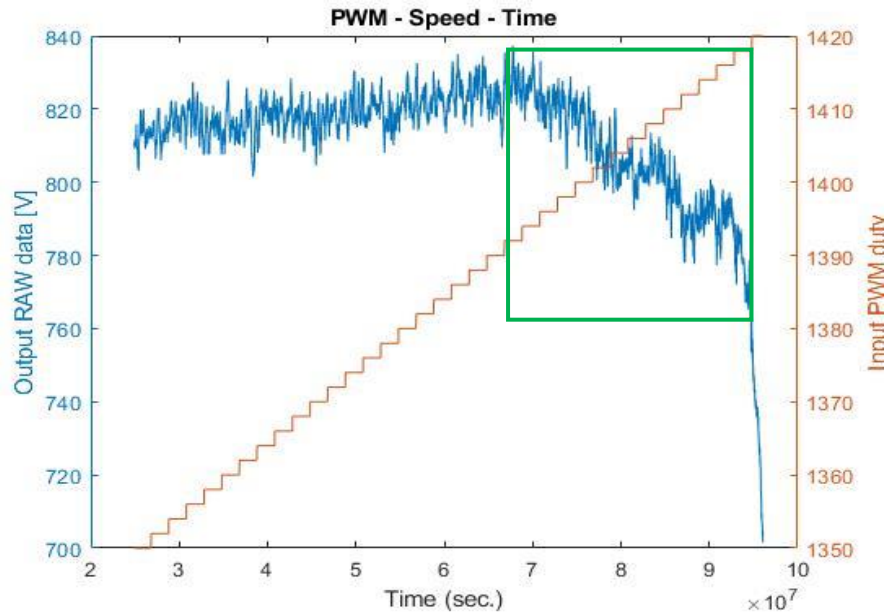


FIGURE 30. OUTPUT FROM THE ACCELEROMETER, IN GREEN THE ANALYSED PART

With this output the data considered was the one from the moment when the voltage was notoriously changing as that part is when the bar starts to move upwards (Green Square on Figure 31).

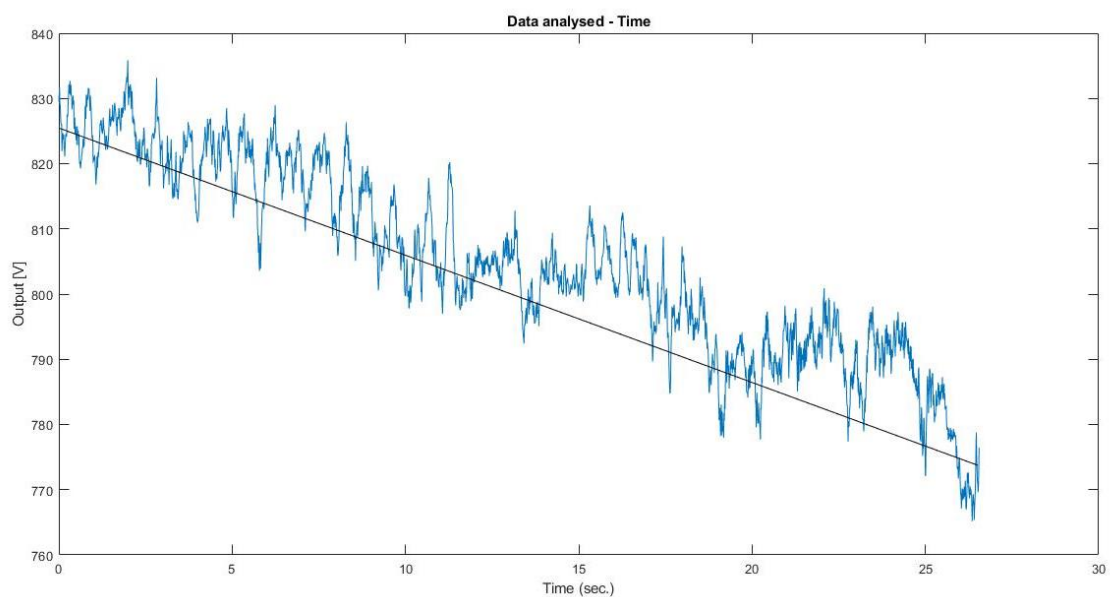


FIGURE 31. CONSIDERED DATA FROM THE ACCELEROMETER

After the data was cut, the linearization was done with different combinations of zeros and poles even that the expected behaviour is to have zero poles and one or two zeros as it is for a PI system. The app “System Identification” from MatLab was used to do it. The following Table 4 is showing the result of the plant:

| |
|--|
| <div><div>G_s =</div><div>From input "u1" to output "y1":</div><div><div>1.328e06 s - 4044</div><div>-----</div><div>s^3 + 245.3 s^2 + 2.315e06 s + 2.181e-07</div></div><div>Name: G_s</div><div>Continuous-time identified transfer function.</div><div>Parameterization:</div><div><div>Number of poles: 3</div><div>Number of zeros: 1</div></div><div>Number of free coefficients: 5</div><div>Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.</div><div>Status:</div><div>Estimated using TFEST on time domain data "mydata".</div><div>Fit to estimation data: 63.34% (simulation focus)</div><div>FPE: 29.99, MSE: 29.8</div></div> |
|--|

TABLE 4. RESULTS FOR THE PLANT OF THE PID CONTROL AFTER ANALYSING THE DATA FROM THE ACCELEROMETER

10.3.3. Replacement of the sensor

In the Figure 32 there is the data when the bar is held on the wood support (left green point), when the propeller starts turning but is still on the support, the output is higher (right green point). When the bar is starting to move upwards (orange points), the reading of the sensor is decreasing rapidly as expected.

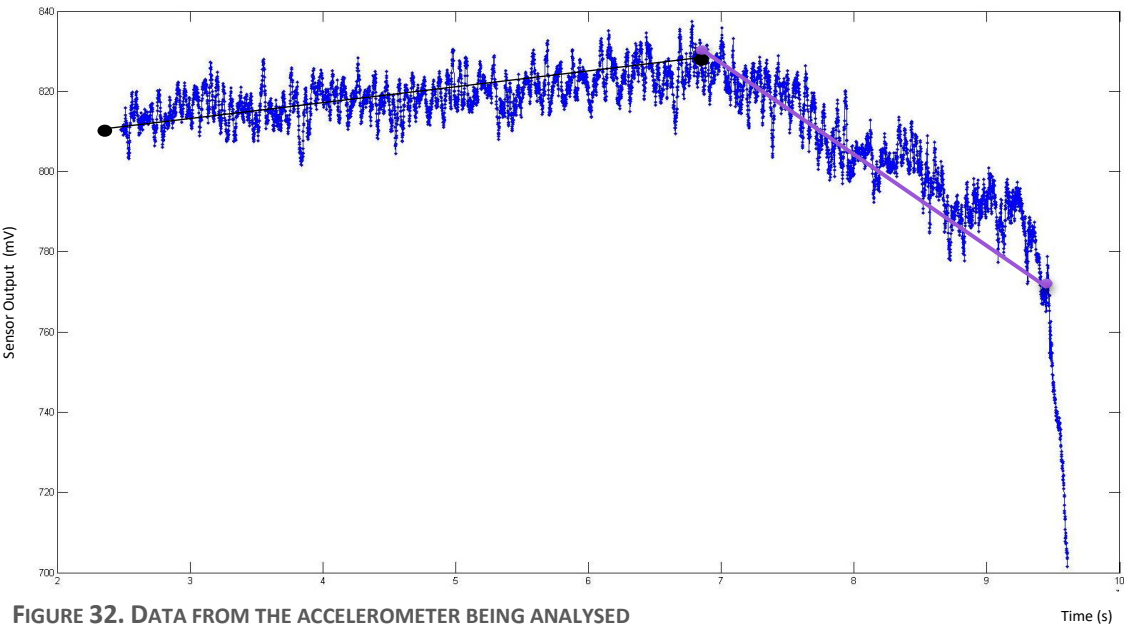


FIGURE 32. DATA FROM THE ACCELEROMETER BEING ANALYSED

The strange behaviour occurs when the bar is still on the support. It is giving different reading values (difference of 20mV as shown in the graph) when the propeller starts turning but is still supported on the base. Therefore, the sensor is not giving a realistic value and cannot be considered for this project.

The final decision of replacement due to the point previously explained had to be taken. There were several possibilities but the final decision was a Distance Measuring Sensor (explained in Chapter 10.4). Also, the linearization of the function was not as precise as desired for the project.

10.4. Distance Measuring Sensor

This sensor was chosen due to its reliability as it is giving the output depending on infrared light beams instead of having the instability the sensor generated due to its principle of function with gravity on the accelerometer output. Another reason to choose the sensor was because it was easy to place it in the already existing system. The first option was a potentiometer with a metal stick connecting directly the bar and the potentiometer. This option was finally discarded because the needed accuracy of the potentiometer was too high, the unavailability in the laboratory and physically it was more complex.

Secondly, to explain the tests carried out, Table 5 shows the pins of the distance sensor and its functions. The sensor is connected like the accelerometer (Figure 29), a condenser was added as mentioned in the specifications to avoid damaging the sensor (Photo 17) and the sensor while Photo 18 is showing how the sensor was positioned.

TABLE 5. DISTANCE SENSOR PINS AND FUNCTIONS

| <i>Pin Name</i> | <i>Function</i> | <i>Name</i> | <i>Connection to the breadboard</i> |
|-----------------|-------------------------|-----------------|-------------------------------------|
| 1 | Output Terminal Voltage | V _o | Arduino (A0) – Analog Read Pin |
| 2 | Ground | GND | Ground |
| 3 | Supply Voltage | V _{CC} | Condenser + Supply |

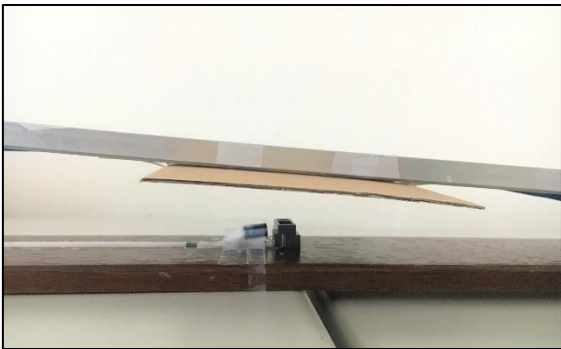


PHOTO 17. ZOOM IN THE DISTANCE SENSOR AND CONDENSER

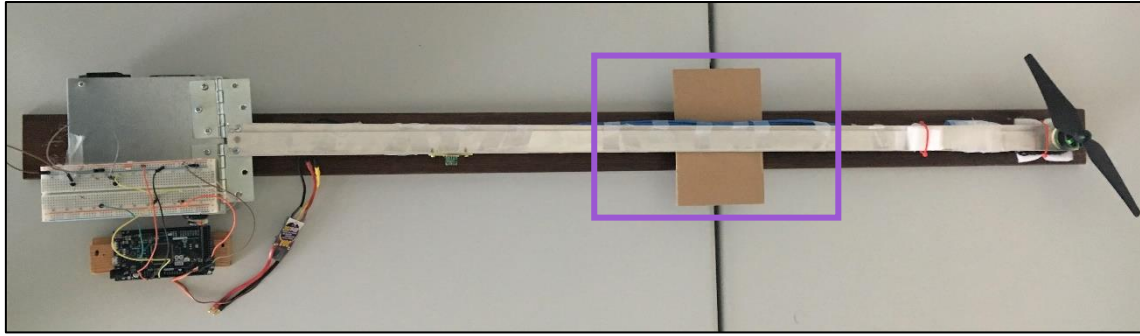


PHOTO 18. PLANT OF THE SYSTEM INDICATING THE DISTANCE SENSOR

10.4.1. Test 1: Performing the previous tests using the output of the distance sensor

Oppositely to the accelerometer, the first tests were showing realistic data without any perturbation caused by the movement of the motor, this can be observed in the Figure 33. After the data analysed was valid, the next steps are explained in the subchapter 10.4.2.

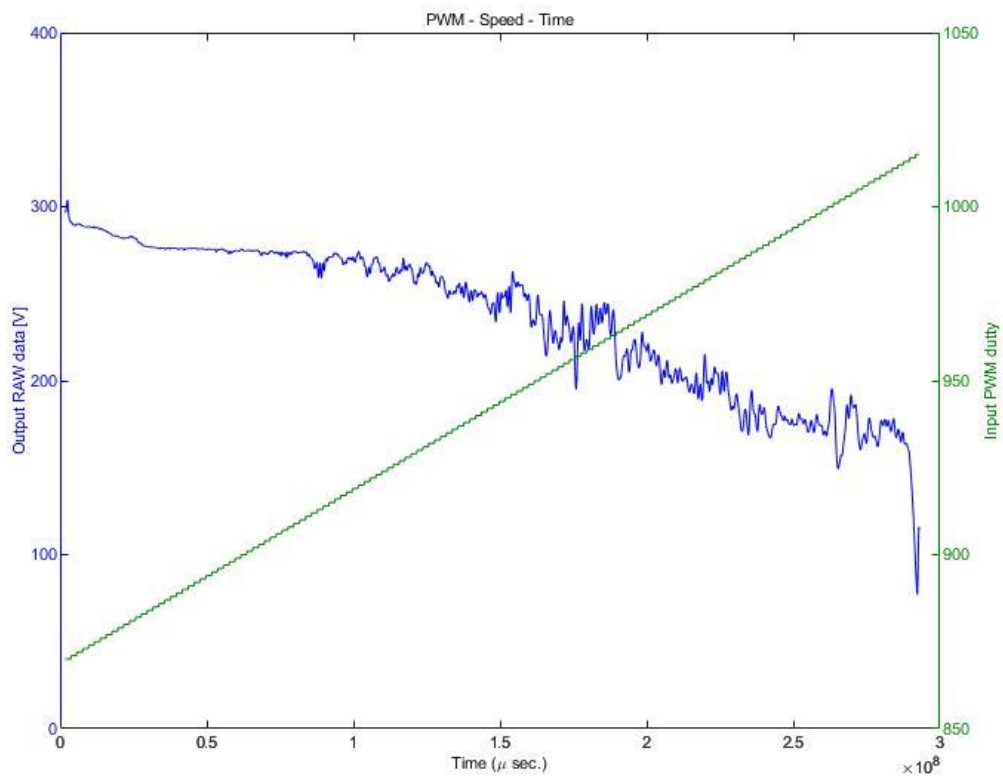


FIGURE 33. DATA TAKEN FROM THE DISTANCE SENSOR

10.4.2. Test 2: Analysing the output of the distance sensor

After analysing different data received, it could be confirmed the behaviour of the system was correct. The next step was taking a reference, in this case the reference chosen was when the bar was being hold by the wood support. This came with the idea of visually changing the graph, as when the sensor was further, the voltage output was lower and this was shown in the graph as a decreasing point. For understanding visually better the information given by the graphs and taking the reference value, a function was prepared with Arduino to show the voltage as an increment (Figure 34). To finally store and import to MatLab.

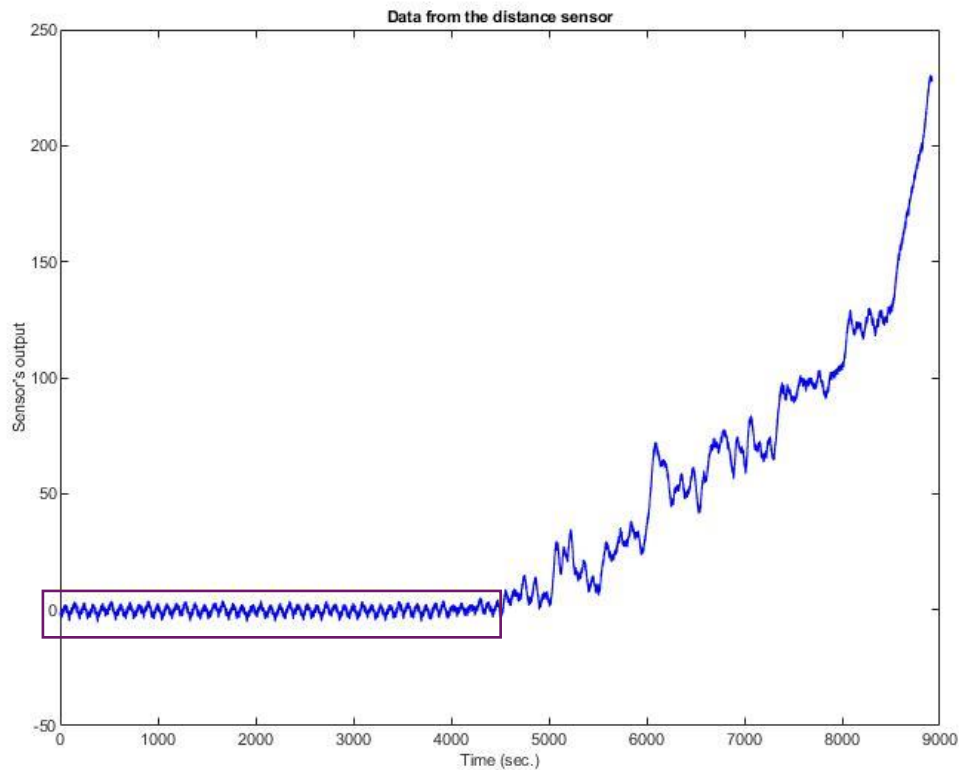


FIGURE 34. DATA OF THE DISTANCE SENSOR WITH THE REFERENCE POINT

10.5. Mathematical model: Linearization of the transfer function

The linearization of the function can be done using several tools in MatLab, in this project is done by using the 'System Identification Tool' which is a special package that can be purchased with the MatLab software. In this system it was considered the PWM (also called '*duty*' during the analysis of data) as input and the output of the sensor ('*dist*' or '*distance*' during the analysis of data) as output for the system. After importing the data it is necessary to choose how many number of poles and zeros the system is going to have, after testing different combinations in the following steps the final results are shown. Photos 19 and 20 are showing the steps taken. The period is 0.01s.

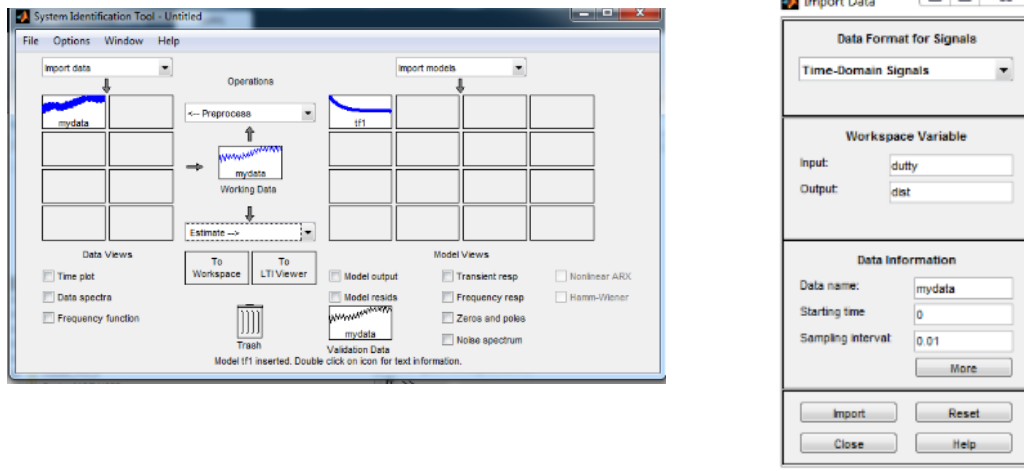


PHOTO 19. IMPORT DATA TO THE 'SYSTEM IDENTIFICATION TOOL'

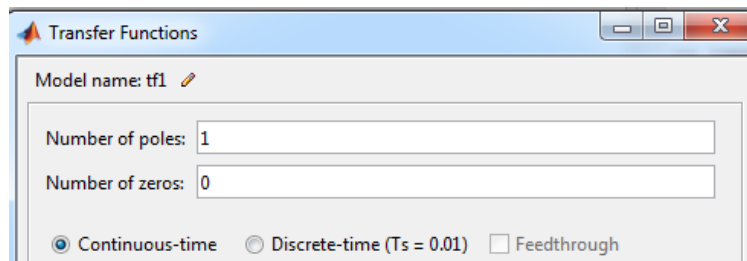


PHOTO 20. 'SYSTEM IDENTIFICATION TOOL' POLES AND ZEROS

The tool final result is showing the information of the plant:

$G_s =$

From input "u1" to output "y1":

0.003519

$s + 7.138e-12$

Name: tf1

Continuous-time identified transfer function.

Parameterization:

Number of poles: 1 Number of zeros: 0

Number of free coefficients: 2

Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
 Estimated using TFEST on time domain data "mydata".
 Fit to estimation data: **76.95%** (simulation focus)
 FPE: 76.87, MSE: 76.77

As this data is in continuous-time domain, a 'Zero-Order Holder' step was added:

```
G_z = c2d(G_s,0.01,'ZOH')
```

After this step, the function is converted to discrete-time domain:

G_z =

From input "u1" to output "y1":

3.519e-05 z^-1

1 - z^-1

Sample time: 0.01 seconds
 Discrete-time identified transfer function.

Parameterization:

Number of poles: 1 Number of zeros: 1

Number of free coefficients: 3

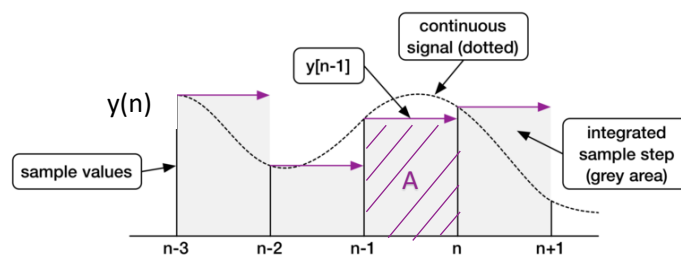
Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Created by direct construction or transformation. Not estimated.

The data is following the Forward Euler formula. There are different numerical methods that would be used to analyse the system and obtain the Transfer Function. A general principle to derive numerical methods is to "discretize" the parts of the function like derivatives, integrals, etc.

Forward Euler method is following the equation in the continuous time domain:



$$A = T \cdot f(n_T - T)$$

$$y(n_T) = y(n_T - T) + T \cdot f(n_T - T)$$

From this formula in continuous-time we operate it to switch to discrete-time:

$$Y(z) = Y(z) z^{-1} + T \cdot F(z) z^{-1}$$

$$Y(z)(1 - z^{-1}) = T \cdot F(z) z^{-1}$$

$$\frac{Y(z)}{F(z)} = \frac{T \cdot z^{-1}}{1 - z^{-1}} = \frac{T}{z - 1} \simeq \frac{1}{s}$$

11. Implemented controller

11.1. Characteristics of the system

The static characteristic of a system is the relation between the input and output in steady state. In this project the input of the system is considered to be the thrust force generated by the propeller (*duty/PWM*) and the output is the position of the system (*distance/sensor's output*), expressed using the voltage output of the sensor. The general is following the next Figure 35:

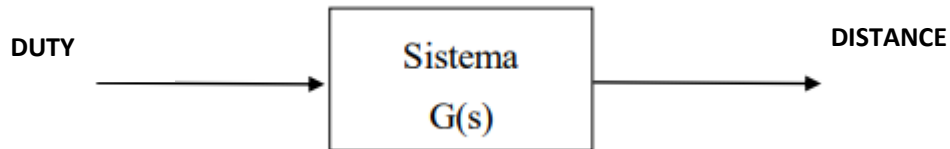


FIGURE 35. GENERAL SCHEME OF THE SYSTEM

The given input of the system is the force needed by the system so it can reach a certain position. This force is depending on the shape of the propeller, speed of rotation and air density, but to obtain the input required by the system, it is needed to know the relation between the force and the signal of control, this is done by using as explained in the Chapter 10.5.

11.2. System designed

The control of the system implemented is done by using a close-loop PI controller. It is following the block diagram presented in the Figure 36:

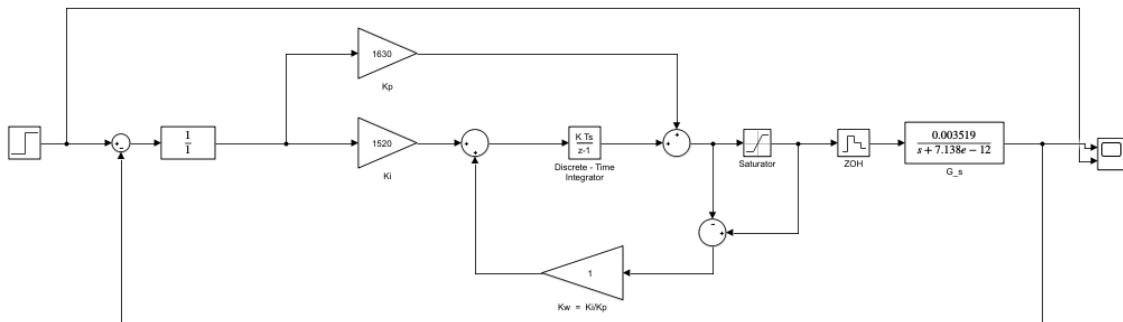


FIGURE 36. SYSTEM IMPLEMENTED WITH THE MATLAB VALUES

The proposed system is due to the PID tuner mentioned in the Chapter 8.8.2, after obtaining the discrete-time transfer function (Chapter 10.5), the function is imported to the PID Tuner Tool and the data is analysed, Figure 37:

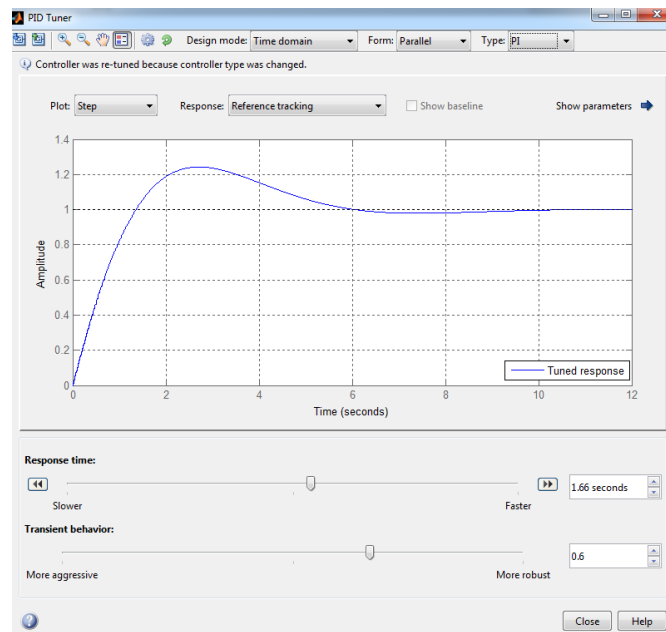


FIGURE 37. PID TUNER TOOL INTERFACE WITH THE IMPORTED DATA

The type of control and parameters of 'Time Response' and 'Transient Behaviour' can be configured to obtain the desired response, Figure 38 is showing the response and values for the K_I , K_P and K_D constants given by the PID Tuner Tool:

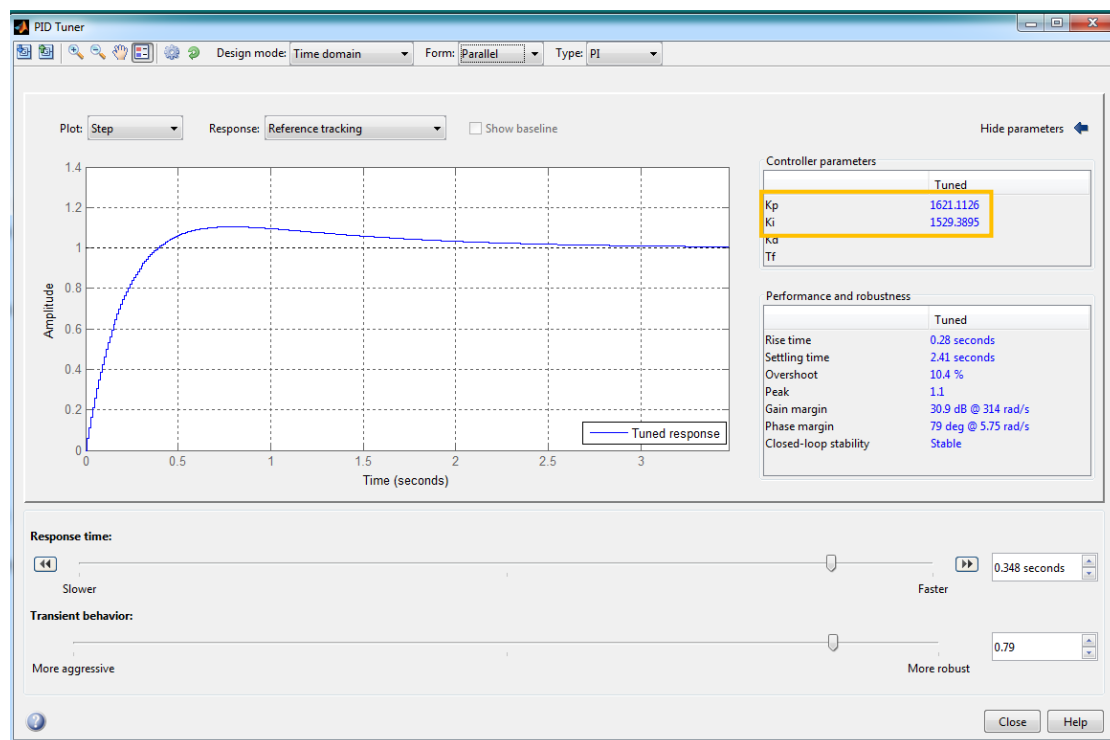


FIGURE 38. PID TUNER RESPONSE TO THE SYSTEM IMPLEMENTED

Depending on the system chosen, the variables K_I , K_P and K_D are given to easily be implemented on the controller. For the system implemented MatLab is giving the following K_I and K_P parameters:

$$\begin{aligned}
 K_I &= 1530 \\
 K_P &= 1620 \\
 K_W &= 0.95 \\
 G_S &= \frac{0.003519}{s + 7.138 \cdot 10^{-12}}
 \end{aligned}$$

Also given by the PID tuner is the final result of the controller function:

C =

$$\mathbf{Kp + Ki * \frac{Ts}{z-1}}$$

with **Kp = 1.62e+03**, **Ki = 1.53e+03**, Ts = 0.01

This function is the one implemented for the controller in Arduino together with the error:

```

void Controller()
{
    error = ref - (ref_data-vm);          //error
    w = Control_Signal - Control_Signal_Sat; //anti-windup

    Control_Signal = error * Kp +
        (-Kp + Ki * T) * error_1 +
        Control_Signal_1 - Kw*T/2 * (w+wAnt); //anti-windup

    Control_Signal_1 = Control_Signal;
    error_1 = error;
    wAnt = w;

    // Saturator
    Control_Signal_Sat = Control_Signal;
    if(Control_Signal_Sat > umax) Control_Signal_Sat = umax; //1020
    if(Control_Signal_Sat < umin) Control_Signal_Sat = umin; //960

    PWM_Val = Control_Signal_Sat;
}

```

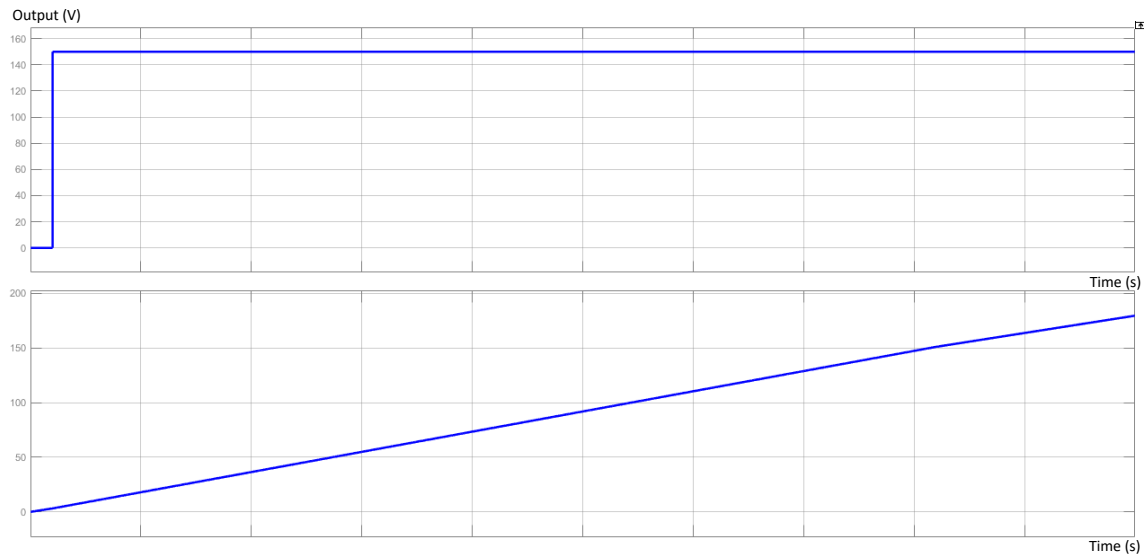


FIGURE 39. OUTPUT GIVEN USING MATLAB VALUES (AT THE TOP IS THE STEP INTRODUCED, BELOW IS THE OUTPUT OF THE SYSTEM)

Figure 39 is showing the output of the system using the variables given by the program. For this project the results given by MatLab were considered firstly and implemented in the real prototype. The desired control of the system was not achieved and the system was not responding as a controller. Using the fundamental method of Trial and Error with the real system but maintaining the transfer function, the final values implemented in the real prototype are:

$$\begin{aligned}
 K_I &= 1.1 \\
 K_P &= 0.1 \\
 K_W &= 11 \\
 G_S &= \frac{0.003519}{s + 7.138 \cdot 10^{-12}}
 \end{aligned}$$

After achieving the requirements of the control, the system was analysed by using Simulink. This is done by using one step and implementing the scheme of the Figure 40. The results are shown in the Figure 41:

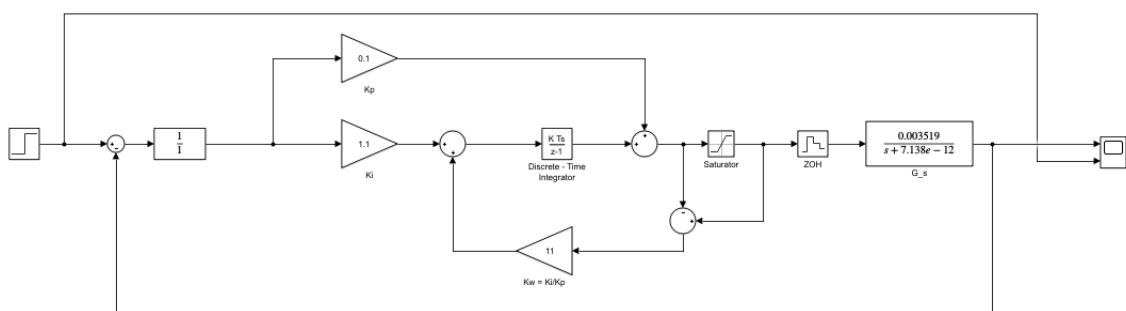


FIGURE 40. IMPLEMENTED PID SYSTEM IN THE REAL PROTOTYPE

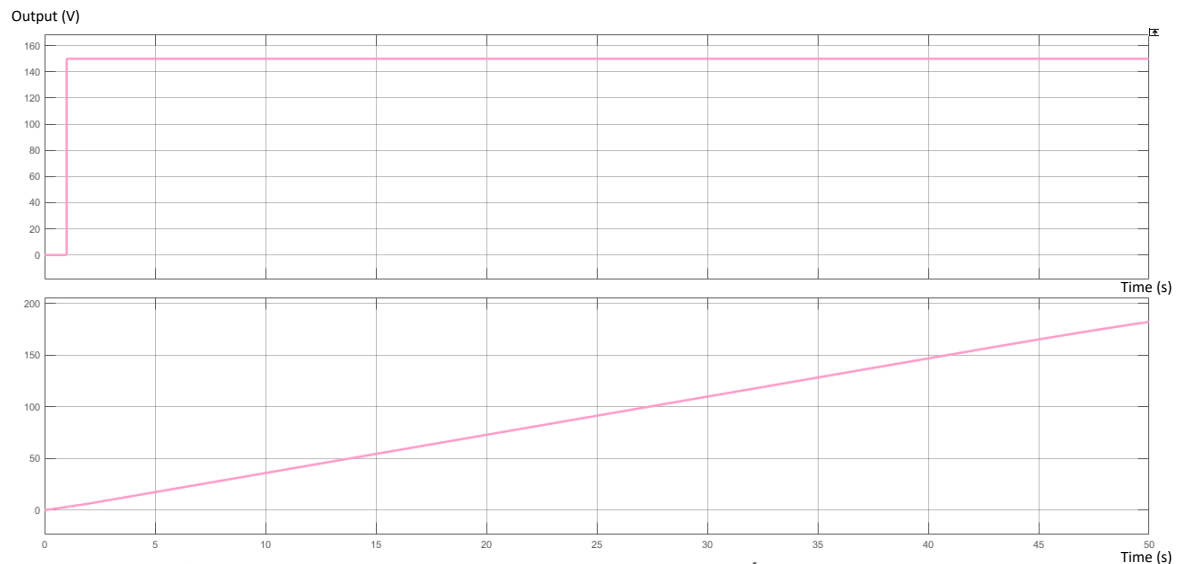


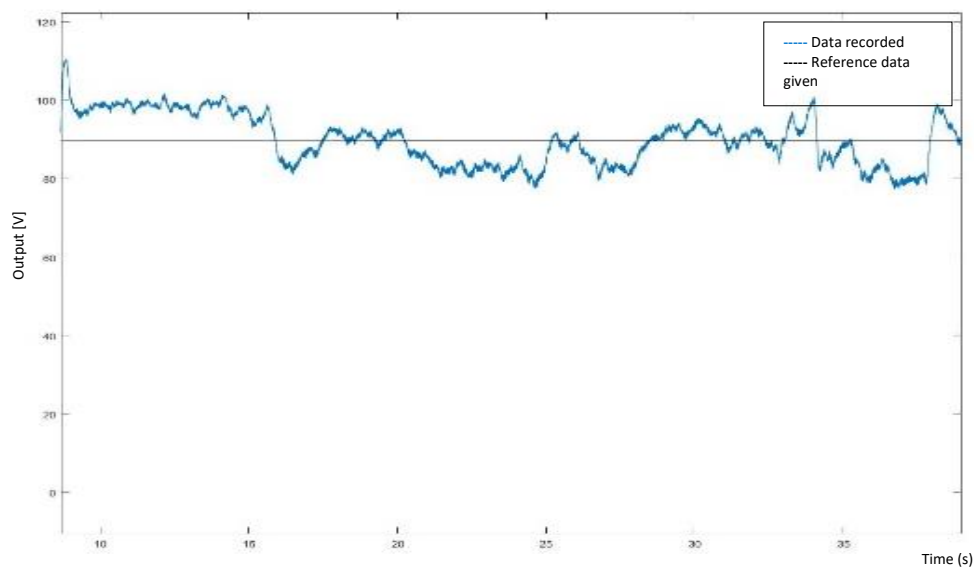
FIGURE 41. SIMULINK RESPONSE FOR THE REAL IMPLEMENTED SYSTEM (AT THE TOP IS THE STEP INTRODUCED, BELOW IS THE OUTPUT OF THE SYSTEM)

As the previous Figure 41 is not showing a realistic output value for our system, the simulated system was improved and it can be found in the Chapter 11.3.

11.3. Improved controller

The results of computing and calculating the Transfer Function were done by recording the data using the PWM signal given to the ESC continuously increasing and the sensor's output was being recorded to obtain the plant of the system. Another way of obtaining the function is to perform the tests by setting a constant PWM/duty value and reading the output of the sensor.

In order to achieve a new transfer function based on a constant system, the PWM was set to a number¹² and the values given by the sensor's output were recorded. After this, the data was imported and analysed on MatLab as explained in the Chapter 10.5. Figure 42 is showing the data recorded.



¹² In this case the raw number was 1010, where the bar was standing horizontally from the wood support and the output of the sensor was known to be 100 from the reference point.

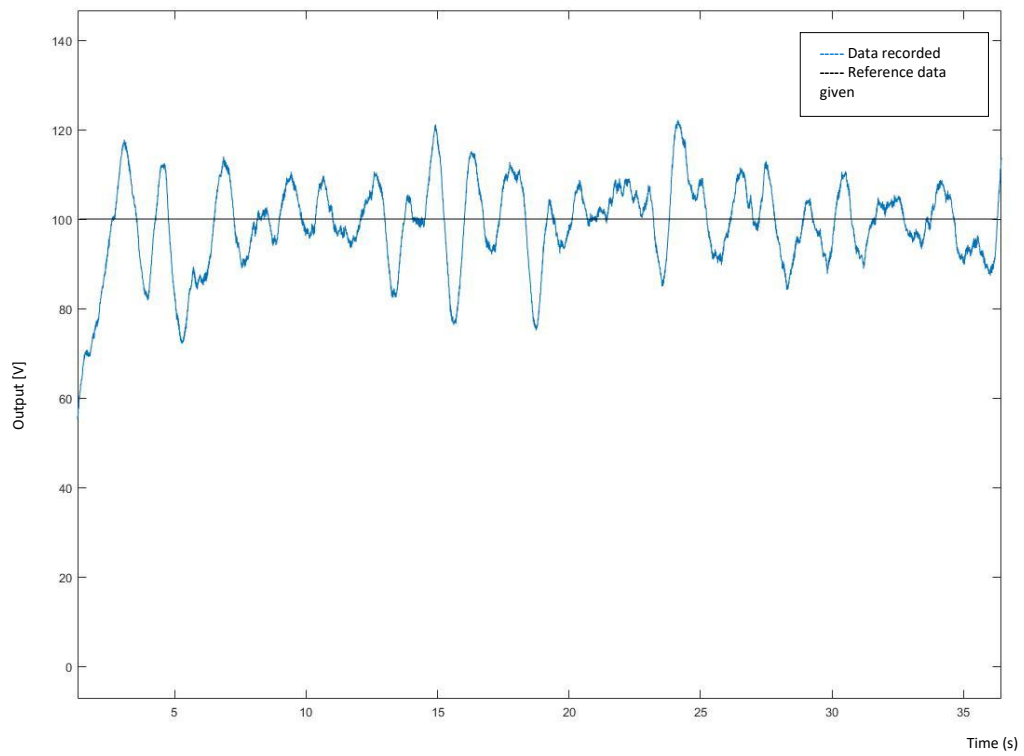
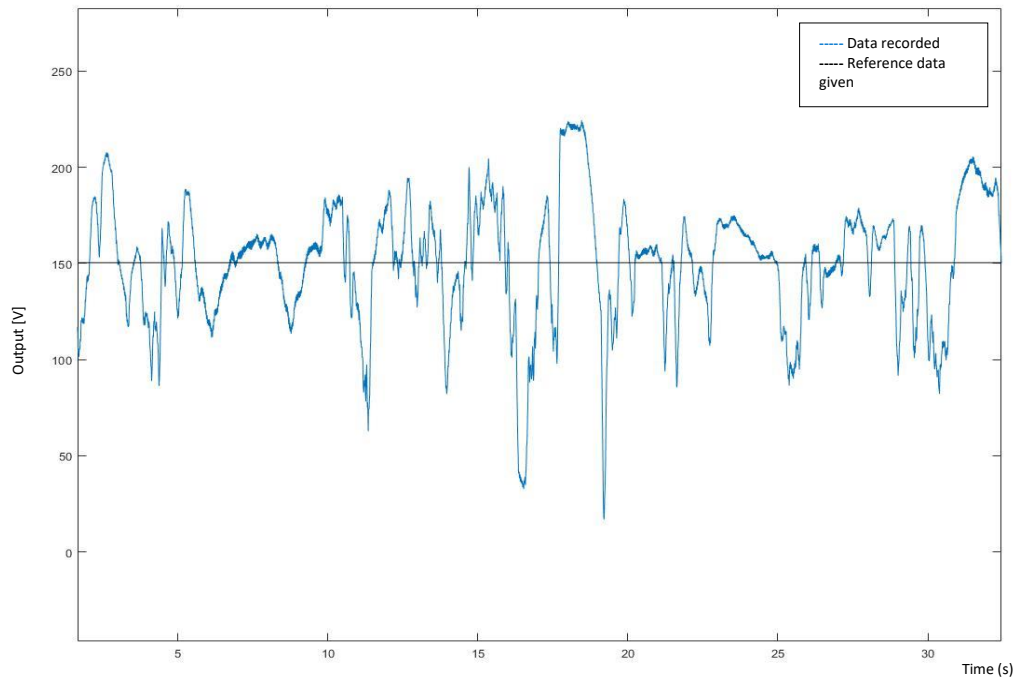


FIGURE 42. DATA RECORDED FOR A CONSTANT DUTY WITH DIFFERENT REFERENCE DISTANCES 85 V [TOP], 150V [MIDDLE], 100V [BOTTOM]

The result of the 'System Identification Tool' for the plant in continuous-time is:

G_s =

From input "u1" to output "G_s":

1.33

s² + 1.791 s + 13.67

Name: G_s

Continuous-time identified transfer function.

Parameterization:

Number of poles: 2 Number of zeros: 0

Number of free coefficients: 3

Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using TFEST on time domain data "mydata".

Fit to estimation data: **63.02%** (simulation focus)

FPE: 98.78, MSE: 98.03

The result after applying the 'Zero-Order Holder' function to transform the result to discrete-time is:

G_z =

From input "u1" to output "G_z":

6.997e-05

1 - 1.976 z⁻¹ + 0.9764 z⁻²

Name: G_z

Sample time: 0.01 seconds

Discrete-time identified transfer function.

Parameterization:

Number of poles: 2 Number of zeros: 0

Number of free coefficients: 3

Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:

Estimated using TFEST on time domain data "mydata".

Fit to estimation data: 42% (simulation focus)

FPE: 243, MSE: 241.2

Introducing the last function (G_z) to the PID Tuner tool, the first idea was to implement a PID controller, the output of this is shown in the Figure 43, it can be seen the function is never stabilized and therefore, the controller type had to be changed.

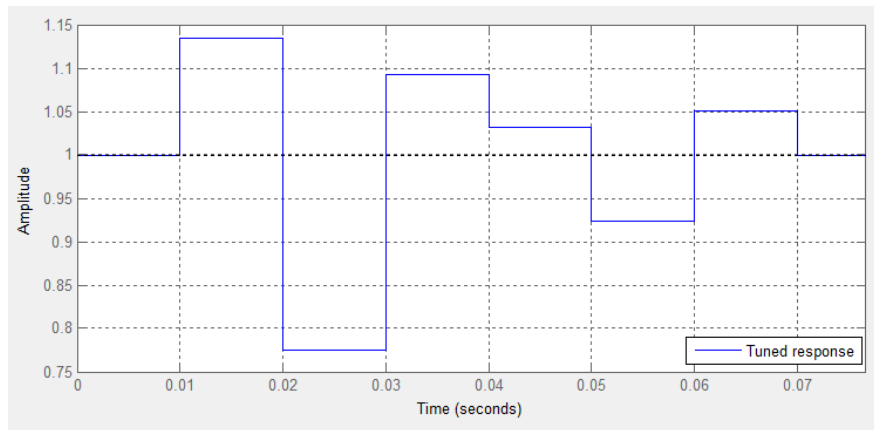


FIGURE 43. PID TUNER OUTPUT TO IMPLEMENT A PID CONTROLLER FOR THE OPEN-LOOP SYSTEM

As the previous control was a PI, it was again retested for a PI giving the output shown in the Figure 44:

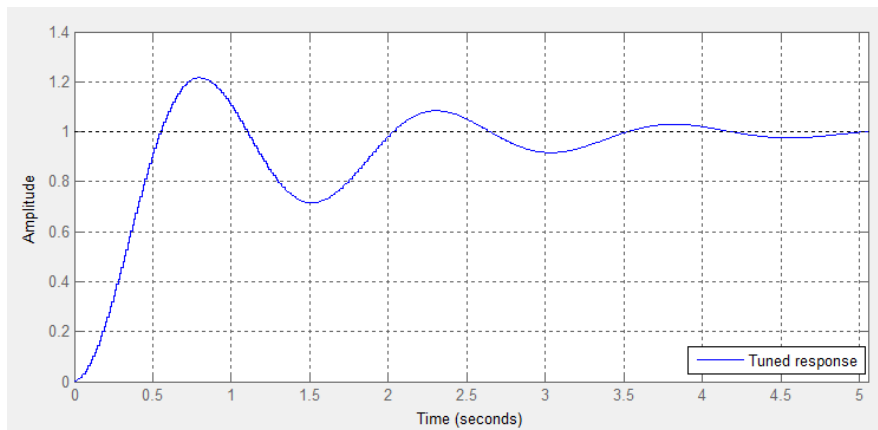


FIGURE 44. PID TUNER OUTPUT TO IMPLEMENT A PI CONTROLLER FOR THE OPEN-LOOP SYSTEM

Following Figure 44, the formula given by the output is:

C =

$$K_p + K_i * \frac{T_s}{z-1}$$

with **K_p = 16.8**, **K_i = 24.9**, **T_s = 0.01**

Sample time: 0.01 seconds

Discrete-time PI controller in parallel form.

Using a **Response Time** of **0.506 seconds** and a **Transient Behaviour** of **0.3**.

Finally, the values for the controller were introduced to the Simulink simulator following the schema of the Figure 45 with the new G_s function:

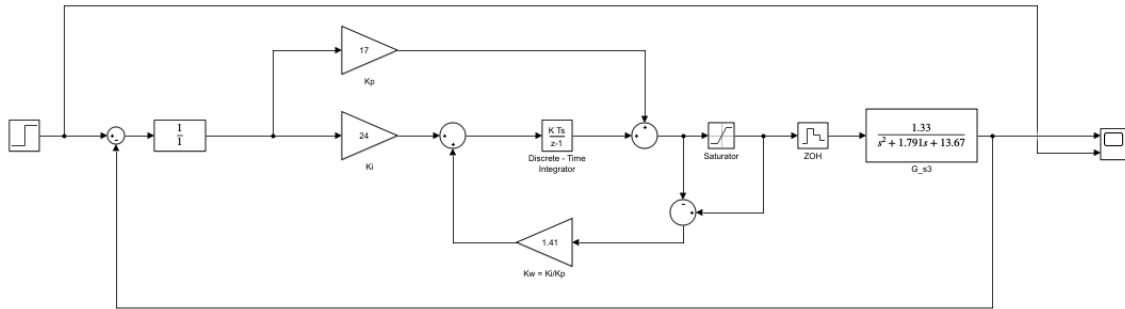


FIGURE 45. IMPROVED PID CONTROLLER SCHEMA ON SIMULINK

After analysing the results, and performing tests with different values for the controller, the final implemented PI on the simulations was using the following parameters:

$$\begin{aligned}
 K_I &= 24 \\
 K_P &= 17 \\
 K_W &= 1.41 \\
 G_S &= \frac{1.33}{s^2 + 1.791s + 13.67}
 \end{aligned}$$

With these values the output simulation was performing the behaviour shown in the Figure 46:

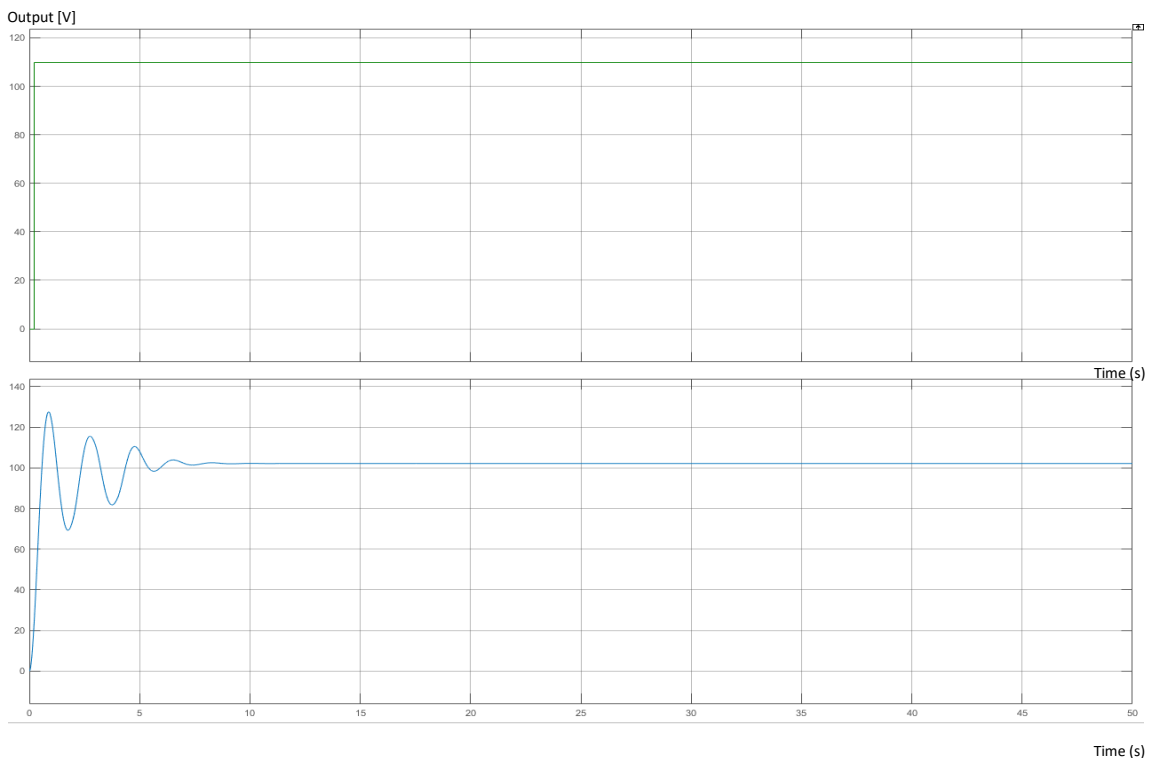


FIGURE 46. FINAL RESULTS FOR THE IMPROVED PI CONTROLLER (AT THE TOP IS THE STEP INTRODUCED, BELOW IS THE OUTPUT OF THE SYSTEM)

In the following Table 6, it is shown a comparison between the values of the PIs tested and analysed on Simulink and MatLab. Below the table in the Figure 47 and 48, the comparison between the 'step' and the output results is shown:

TABLE 6. COMPARISON BETWEEN SYSTEMS

| | <i>MatLab Parameters</i> | <i>Implemented Parameters</i> | <i>Simulated system with MatLab and Simulink</i> |
|-------|---|---|---|
| K_P | 1630 | 0.1 | 17 |
| K_I | 1520 | 1.1 | 24 |
| K_D | - | - | - |
| K_W | 0.95 | 11 | 1.41 |
| G_S | $\frac{0.003519}{s + 7.138 \cdot 10^{-12}}$ | $\frac{0.003519}{s + 7.138 \cdot 10^{-12}}$ | $\frac{1.33}{s^2 + 1.791s + 13.67}$ |
| G_Z | $\frac{3.519 \cdot 10^{-5} z^{-1}}{1 - z^{-1}}$ | $\frac{3.519 \cdot 10^{-5} z^{-1}}{1 - z^{-1}}$ | $\frac{6.997 \cdot 10^{-5}}{1 - 1.976z^{-1} + 0.9764 z^{-2}}$ |

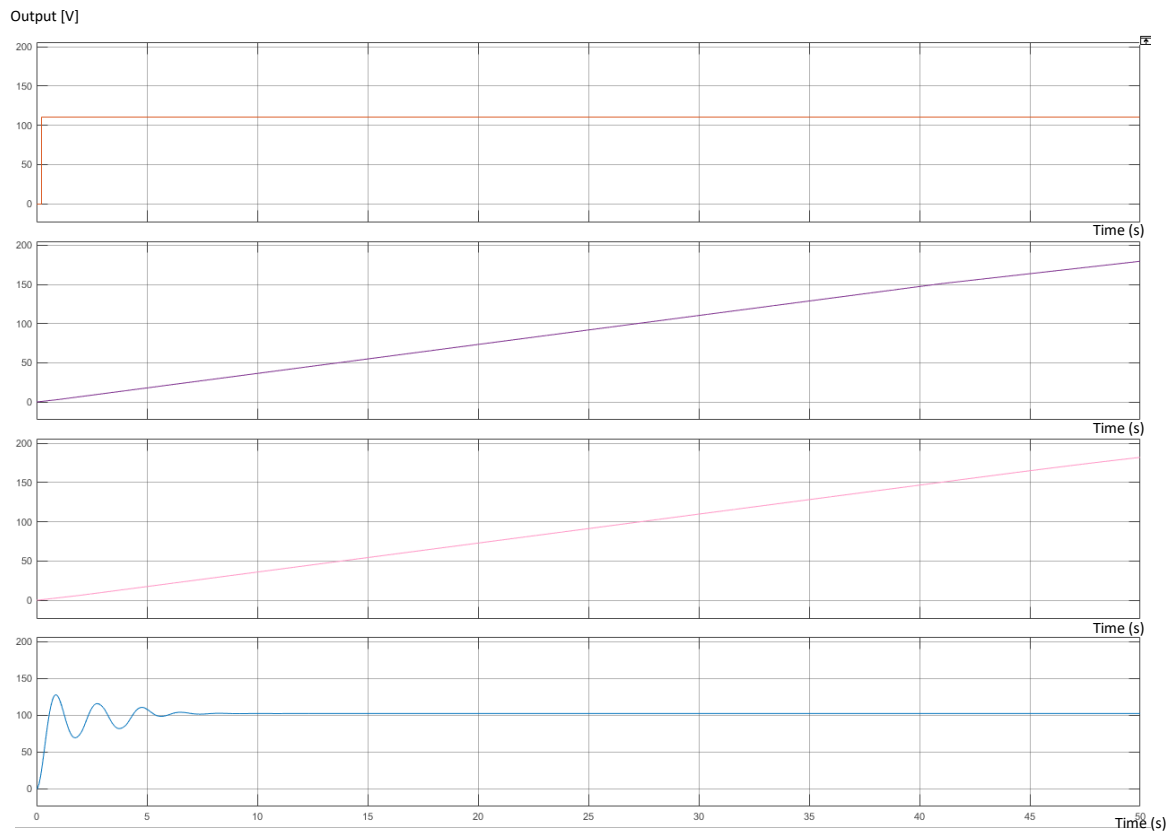


FIGURE 47. COMPARISON BETWEEN THE SYSTEM'S OUTPUT ON SIMULINK (FROM TOP TO BOTTOM: THE INPUT STEP, THE OUTPUT FROM THE MatLab SYSTEM, OUTPUT OF THE IMPLEMENTED SYSTEM AND THE IMPROVED SYSTEM ONLY SIMULATED)

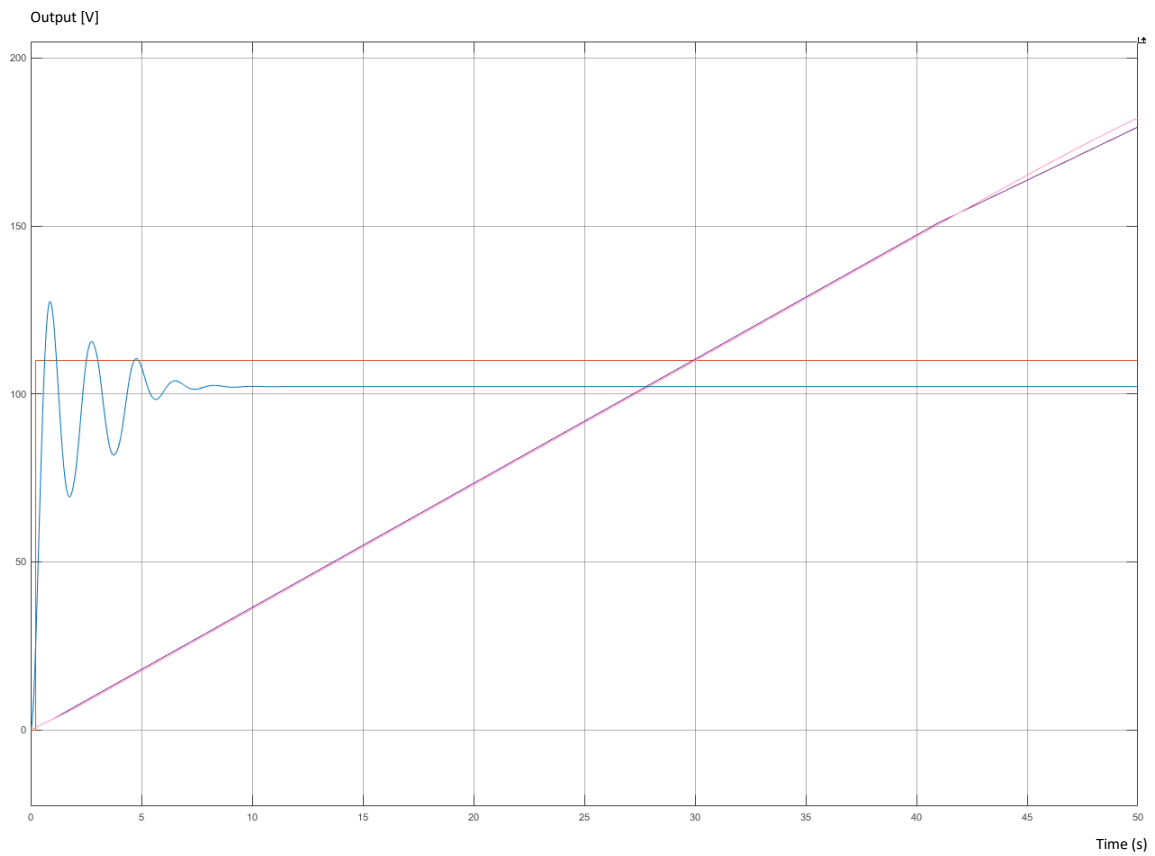


FIGURE 48. COMPARISON SUPERIMPOSED BETWEEN THE SYSTEM'S OUTPUT ON SIMULINK (IN ORANGE THE INPUT STEP, IN PURPLE THE OUTPUT FROM THE MATLAB SYSTEM, IN PINK OUTPUT OF THE IMPLEMENTED SYSTEM AND IN BLUE THE IMPROVED SYSTEM ONLY SIMULATED)

12. Conclusions

In this report it is described the materials and methods used to design and construct a prototype of a motor-propeller-rectangular bar system.

Addressing to the main goal of the project which was being able to construct a controller for an unstable system, it can be considered achieved. It still should be mentioned that even there is a final solution implemented and it is working properly as presented, this could be improved to have a higher control over the system. As a second objective, this report could also be used to be a guide about how to do a PID control using MatLab tools and Arduino.

12.1. Personal assessment

The biggest problem that had to be faced during this project was the behaviour of the ESC. Even the signal generated was correct, in some tests and with no apparent reason for it, the controller was not following the indications given by the PWM signal. This behaviour was tried to be avoided as explained by reconfiguring it but without succeed. The replacement of the ESC did not change the behaviour for the other controller used. This suggested a mistake in the hardware part but after deeply analysing all the possibilities this premise was discarded.

In addition to the main goal, this project has been enriching my personal knowledge and the result of it is satisfactory. The project has been a good opportunity to refresh and improve different knowledge areas such as electronics and programming.

12.2. Future steps on this project

As a future project I would propose to change the ESC controller used by a new one compatible with the motor in order to have a full understanding when interacting with it. After being able to control the ESC the tests performed on chapter 10 and 11 could be repeated to finally have a PID control instead of a PI, as this would be more effective. The formula for the controller would have to be changed on Arduino to implement the derivative part of the PID controller and with this it might be possible to achieve a final result without the anti-windup.

13. References

- Arduino.cc. (sense data). What is Arduino?
- Brahim GASBAOUI, C. A. (24 / December / 2010). Adaptive Fuzzy PI of Double Wheeled Electric Vehicle Drive Controlled by Direct Torque Control. Oran, Algeria.
- Bueno, D. (n.d.). Motor eléctrico brushless: Funcionamiento y características.
- Frick, C. (sense data). Brushless DC Motors Introduction for Next-Generation Missile Actuation Systems Outline.
- Frontz, M. (sense data). Accelerometer.
- José Carlos Gamazo-Real, E. V.-S.-G. (2010, July 19). Position and Speed Control of Brushless DC Motors Using Sensorless Techniques and Application Trends.
- MathWorks. (2019). MatLab.
- SparkFun. (sense data). Accelerometer Basics.

Annex I. Datasheet of the distance measuring sensor

GP2Y0A41SK0F

Distance Measuring Sensor Unit
Measuring distance : 4 to 30 cm
Analog output type



■Description

GP2Y0A41SK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IR-LED (infrared emitting diode) and signal processing circuit. The variety of the reflectivity of the object, the environmental temperature and the operating duration are not influenced easily to the distance detection because of adopting the triangulation method. This device outputs the voltage corresponding to the detection distance. So this sensor can also be used as a proximity sensor.

■Agency approvals/Compliance

1. Compliant with RoHS directive (2002/95/EC)

■Applications

1. Cleaning robot
2. Personal robot
3. Sanitary

■Features

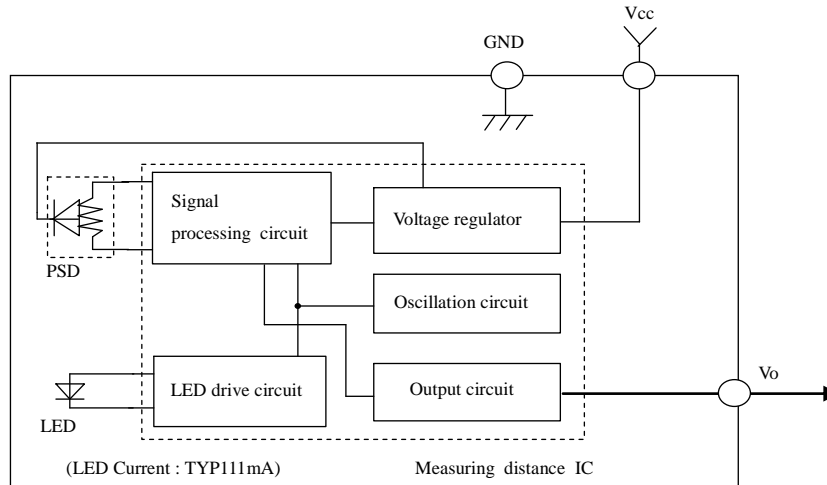
1. Distance measuring sensor is united with PSD, infrared LED and signal processing circuit
2. Short measuring cycle (16.5ms)
3. Distance measuring range : 4 to 30 cm
4. Package size (29.5 × 13.0 × 13.5mm)
5. Analog output type

Notice The content of data sheet is subject to change without prior notice.

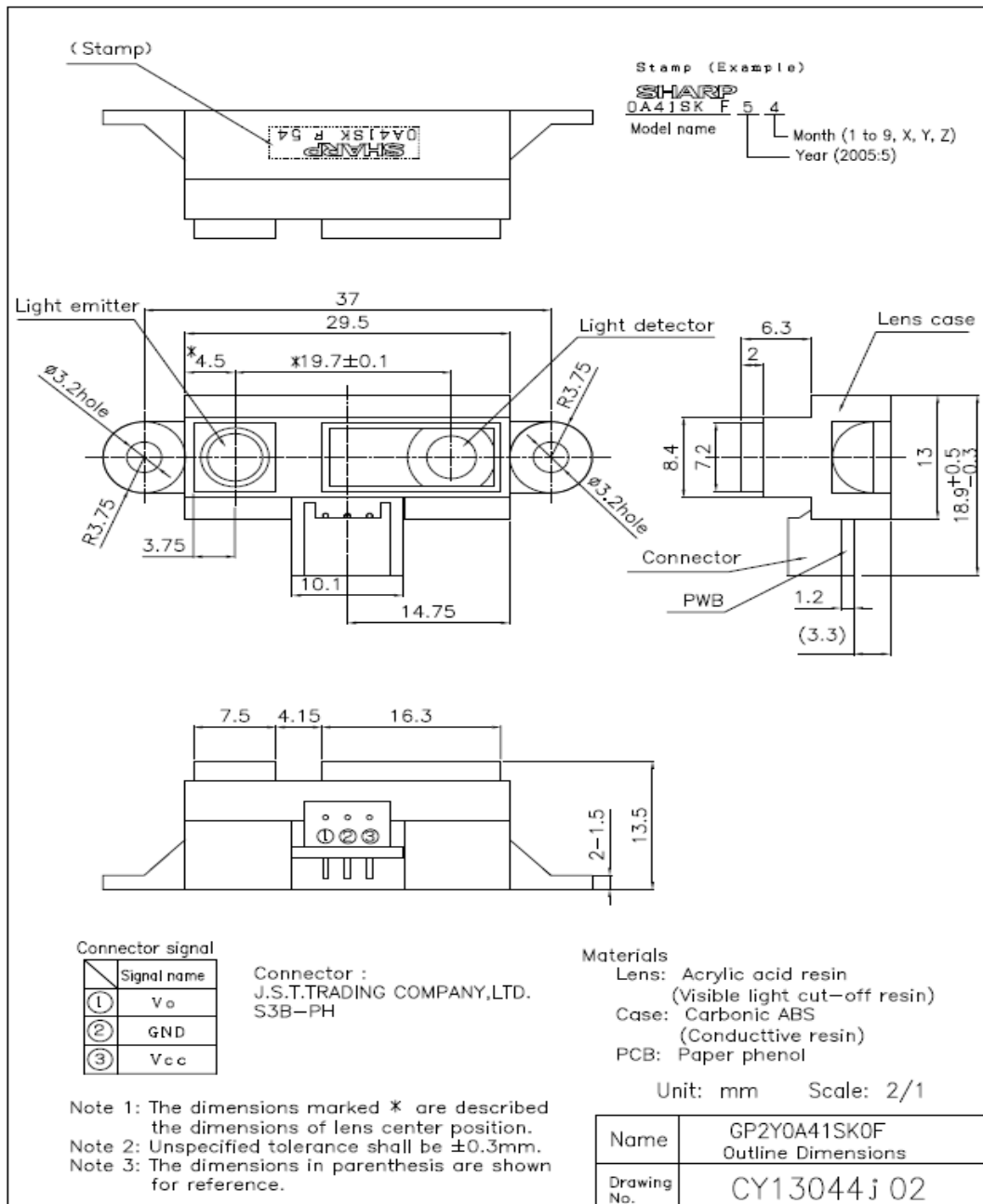
In the absence of confirmation by device specification sheets, SHARP takes no responsibility for any defects that may occur in equipment using any SHARP devices shown in catalogs, data books, etc. Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device.

Sheet No.: OP13008EN

■Schematic



■Outline



■Absolute maximum ratings

(Ta=25°C, Vcc=5V)

| Parameter | Symbol | Ratings | Unit | Remark |
|-------------------------|--------|-----------------|------|--------|
| Supply voltage | Vcc | -0.3 to +7 | V | - |
| Output terminal voltage | Vo | -0.3 to Vcc+0.3 | V | - |
| Operating temperature | Topr | -10 to +60 | °C | - |
| Storage temperature | Tstg | -40 to +70 | °C | - |

■Operating supply voltage

| Symbol | Rating | Unit | Remark |
|--------|------------|------|--------|
| Vcc | 4.5 to 5.5 | V | - |

■Electro-optical Characteristics

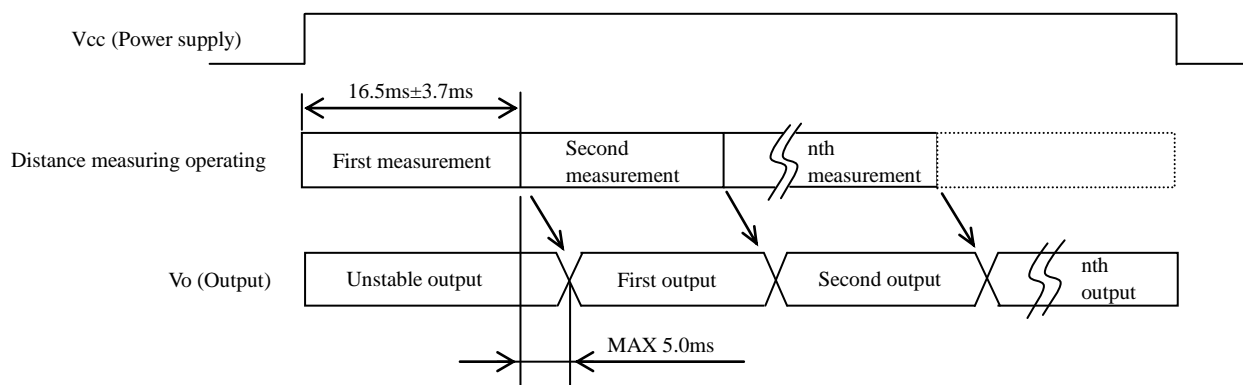
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---------------------------|-------------|--|------|------|------|------|
| Measuring distance range | ΔL | (Note 1) | 4 | - | 30 | Cm |
| Output terminal voltage | Vo | L=30cm (Note 1) | 0.25 | 0.4 | 0.55 | V |
| Output voltage difference | ΔVo | Output change at L change (30cm → 4cm) (Note 1) | 1.95 | 2.25 | 2.55 | V |
| Average supply current | Icc | L=30cm (Note 1) | - | 12 | 22 | mA |

※L : Distance to reflective object

(Note 1) Using reflective object : White paper

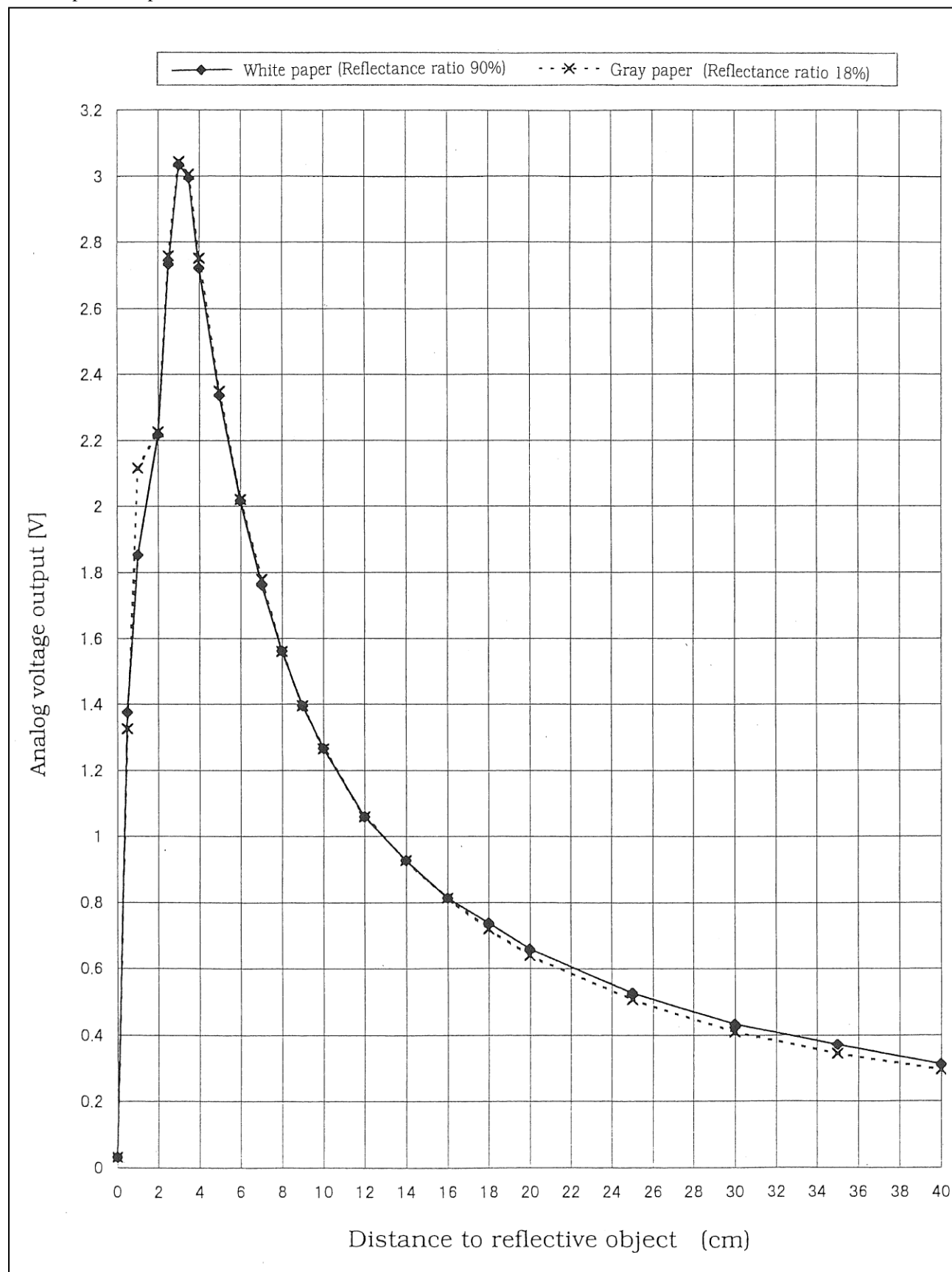
(Made by Kodak Co., Ltd. gray cards R-27 • white face, reflective ratio ; 90%)

■Timing Chart

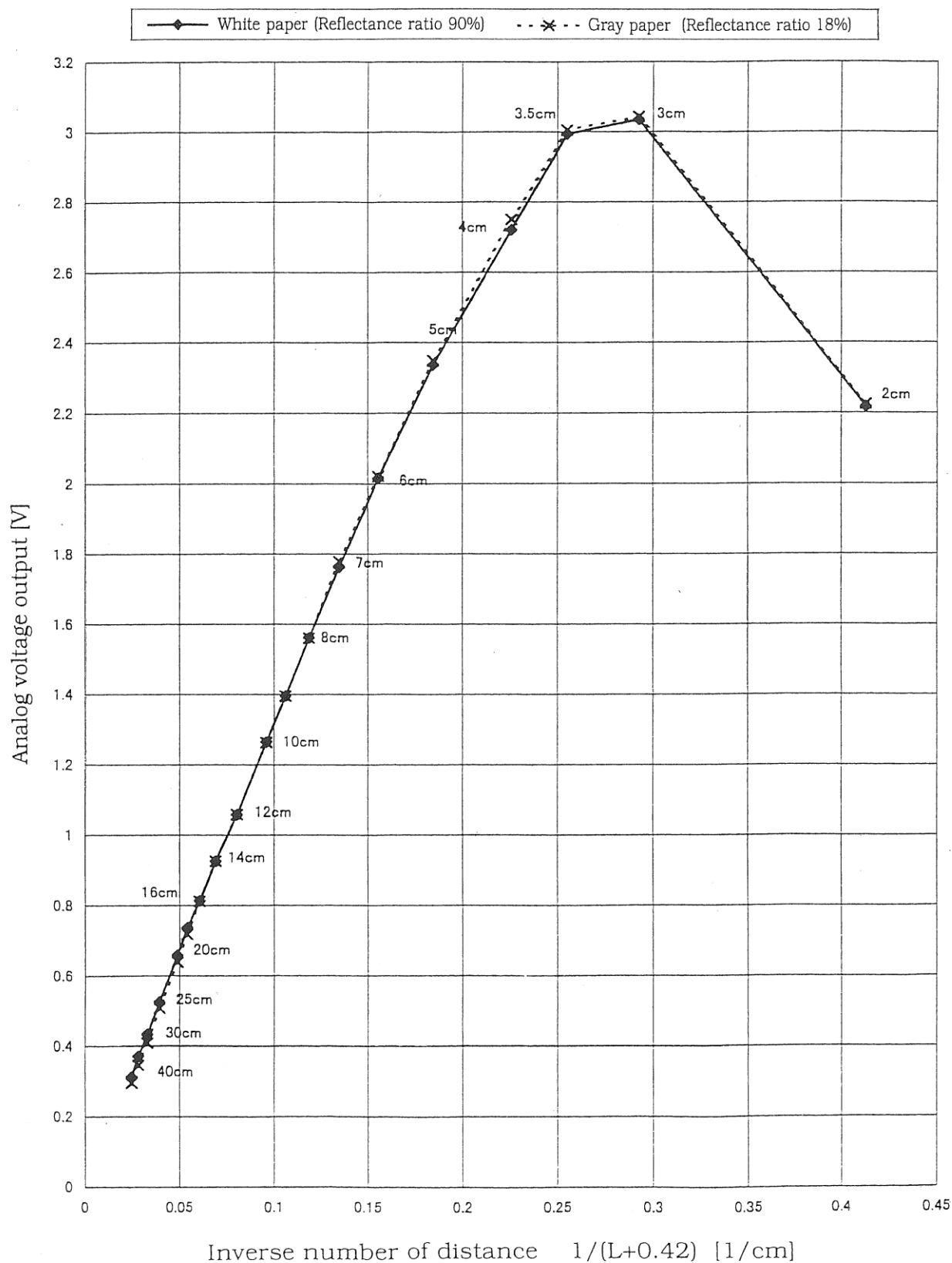


■Supplements

- Example of output distance characteristics



●Example of output distance characteristics with the inverse of distance



- This product shall not contain the following materials.

Also, the following materials shall not be used in the production process for this product.

Materials for ODS : CFC_s, Halon, Carbon tetrachloride 1.1.1-Trichloroethane (Methyl chloroform)

- Product mass : Approx. 3.6g (TYP)

- This product does not contain the chemical materials regulated by RoHS directive.

(Except for the NOT regulated by RoHS directive.)

- Compliance with each regulation

1) The RoHS directive(2002/95/EC)

This product complies with the RoHS directive(2002/95/EC) .

Object substances: mercury, lead (except for lead in high melting temperature type solders^{*1} and glass of electronic components), cadmium, hexavalent chromium, polybrominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE)

*1 : i.e. tin-lead solder alloys containing more than 85% lead

2) Content of six substances specified in Management Methods for Control of Pollution Caused by Electronic Information Products Regulation (Chinese : 电子信息产品污染控制管理办法).

| Category | Toxic and hazardous substances | | | | | |
|---------------------------|--------------------------------|--------------|--------------|---|--------------------------------|---------------------------------------|
| | Lead (Pb) | Mercury (Hg) | Cadmium (Cd) | Hexavalent chromium (Cr ⁶⁺) | Polybrominated biphenyls (PBB) | Polybrominated diphenyl ethers (PBDE) |
| Distance Measuring Sensor | * | ✓ | ✓ | ✓ | ✓ | ✓ |

✓ : indicates that the content of the toxic and hazardous substance in all the homogeneous materials of the part is below the concentration limit requirement as described in SJ/T 11363-2006 standard .

* : indicates that the content of the toxic and hazardous substance in at least one homogeneous material of the part exceeds the concentration limit requirement as described in SJ/T 11363-2006 standard.

Lead in high melting temperature type solders (i.e. tin-lead solder alloys containing more than 85% lead) and glass of electronic components (designated by “*” in the above table) are exempt from the RoHS directive (2002/95/EC), because there is no effective way to eliminate or substitute them by present scientific technology.

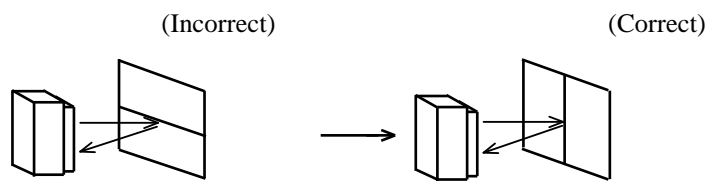
■Notes

[Advice for the optics]

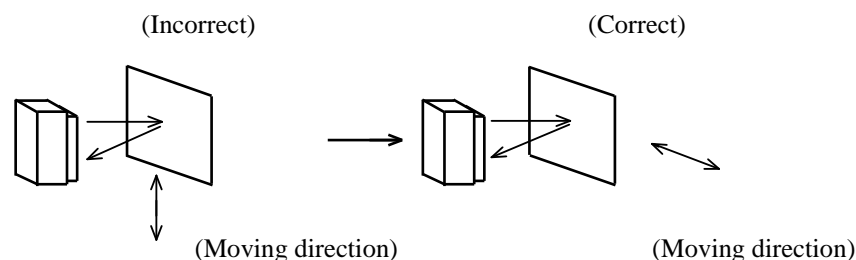
- Lens of this device shall be kept cleanly. There are cases that dust, water or oil and so on deteriorate the characteristics of this device. Please consider in actual application.
- In case that protection is set in front of the emitter and detector portion, the protection cover which has the most efficient transmittance at the emitting wavelength range of LED for this product ($\lambda=870\text{nm}\pm 70\text{nm}$), shall be recommended to use. The face and back of protection cover should be mirror polishing. Also, as there are cases that the characteristics may not be satisfied with according to the distance between the protection cover and this product or the thickness of the protection cover, please use this product after confirming the operation sufficiently in actual application.

[Advice for the characteristics]

- In case that there is an object near to light exits of the sensor between the sensor and the detected object, please use this device after confirming sufficiently what the characteristics of this sensor do not change by the object.
- When the detector surface receive direct light from the sun, tungsten lamp and so on, there are cases that it can not measure the distance exactly. Please consider the design that the detector does not receive direct light from such light source.
- Distance between sensor and mirror reflector can not sometimes measure exactly.
In case of changing the mounting angle of this product, it may measure the distance exactly.
- In case that reflective object has boundary line clearly, there is cases that distance can not measure exactly.
At that time, if direction of boundary line and the line between emitter center and detector center parallels, it is possible to decrease deviation of measuring distance.



- In order to decrease measuring error by moving direction of object, we recommend to mount the sensor like below drawing.

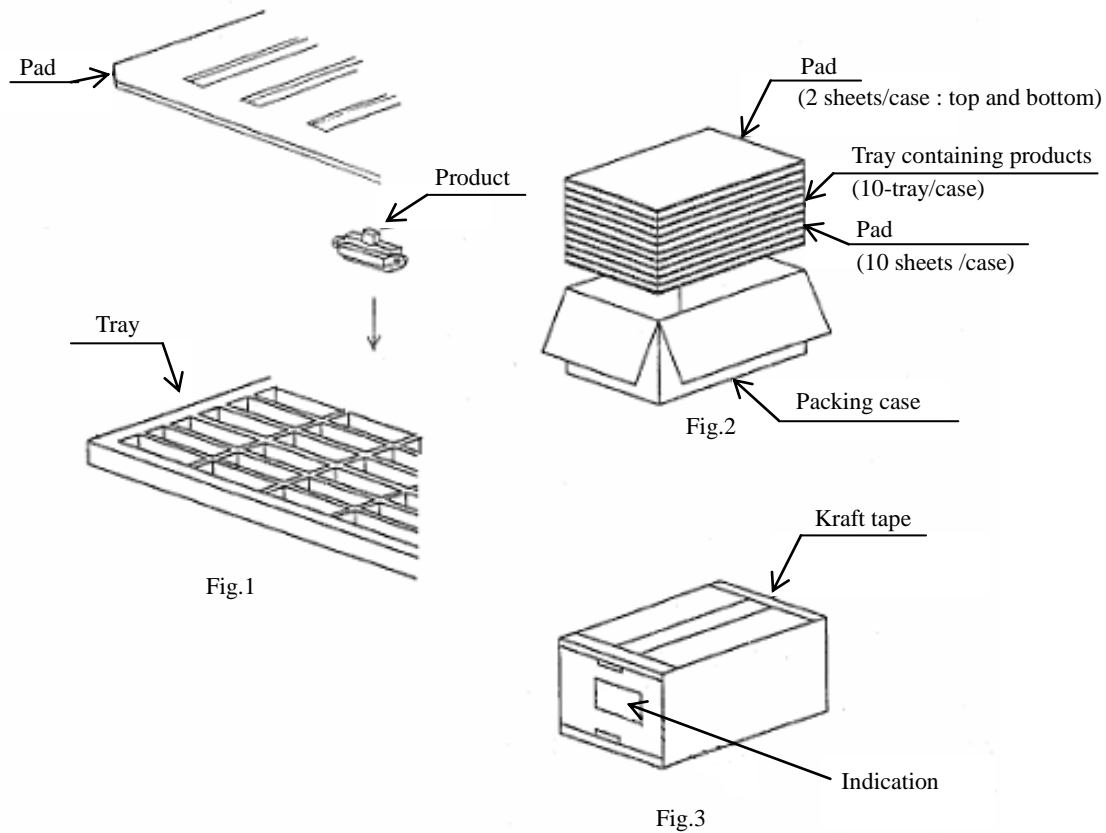


- In order to stabilize power supply line, we recommend to connect a by-pass capacitor of 10 μ F or more between Vcc and GND near this product.

[Notes on handling]

- Please don't do washing. Washing may deteriorate the characteristics of optical system and so on.
Please confirm resistance to chemicals under the actual usage since this product has not been designed against for washing.
- There are some possibilities that the sensor inside the case package with lens may be exposed to the excessive mechanical stress. Please be careful not to cause any excessive pressure on the case package with lens and also on the PCB at the assembly and inserting of the set.

■ Packing specification



1. Packing numbers
 MAX. 100 pieces per tray
 MAX 1000 pieces per case
2. Arranges in 10 stages of trays containing products into the packing case.
 Put pads on their top and bottom.
 Closes the lid of case and seals with kraft tape.
3. Indication items
 The contents of the carton indication conforms to EIAJ C-3 and the following items are indicated.
 Model No., Internal production control name, Quantity, Packing date, Corporate name, Country of origin

■ Important Notices

· The circuit application examples in this publication are provided to explain representative applications of SHARP devices and are not intended to guarantee any circuit design or license any intellectual property rights. SHARP takes no responsibility for any problems related to any intellectual property right of a third party resulting from the use of SHARP's devices.

· Contact SHARP in order to obtain the latest device specification sheets before using any SHARP device. SHARP reserves the right to make changes in the specifications, characteristics, data, materials, structure, and other contents described herein at any time without notice in order to improve design or reliability. Manufacturing locations are also subject to change without notice.

· Observe the following points when using any devices in this publication. SHARP takes no responsibility for damage caused by improper use of the devices which does not meet the conditions and absolute maximum ratings to be used specified in the relevant specification sheet nor meet the following conditions:

(i) The devices in this publication are designed for use in general electronic equipment designs such as:

- Personal computers
- Office automation equipment
- Telecommunication equipment [terminal]
- Test and measurement equipment
- Industrial control
- Audio visual equipment
- Consumer electronics

(ii) Measures such as fail-safe function and redundant design should be taken to ensure reliability and safety when SHARP devices are used for or in connection

with equipment that requires higher reliability such as:

- Transportation control and safety equipment (i.e., aircraft, trains, automobiles, etc.)
- Traffic signals
- Gas leakage sensor breakers
- Alarm equipment
- Various safety devices, etc.

(iii) SHARP devices shall not be used for or in connection with equipment that requires an extremely high level of reliability and safety such as:

- Space applications
- Telecommunication equipment [trunk lines]
- Nuclear power control equipment
- Medical and other life support equipment (e.g., scuba).

· If the SHARP devices listed in this publication fall within the scope of strategic products described in the Foreign Exchange and Foreign Trade Law of Japan, it is necessary to obtain approval to export such SHARP devices.

· This publication is the proprietary product of SHARP and is copyrighted, with all rights reserved. Under the copyright laws, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of SHARP. Express written permission is also required before any use of this publication may be made by a third party.

· Contact and consult with a SHARP representative if there are any questions about the contents of this publication.

Annex II. Datasheet of the accelerometer

±3g, ±9g Three Axis Low-g Micromachined Accelerometer

The MMA7341LC is a low power, low profile capacitive micromachined accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature compensation, self test, and g-Select which allows for the selection between two sensitivities. Zero-g offset and sensitivity are factory set and require no external devices. The MMA7341LC includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

Features

- 3mm x 5mm x 1.0mm LGA-14 Package
- Low Current Consumption: 400 μ A
- Sleep Mode: 3 μ A
- Low Voltage Operation: 2.2 V – 3.6 V
- Selectable Sensitivity (±3g, ±9g)
- Fast Turn On Time (0.5 ms Enable Response Time)
- Self Test for Freefall Detect Diagnosis
- Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- RoHS Compliant
- Environmentally Preferred Product
- Low Cost

Typical Applications

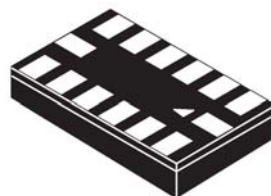
- 3D Gaming: Tilt and Motion Sensing, Event Recorder
- HDD MP3 Player: Freefall Detection
- Laptop PC: Freefall Detection, Anti-Theft
- Cell Phone: Image Stability, Text Scroll, Motion Dialing, eCompass
- Pedometer: Motion Sensing
- PDA: Text Scroll
- Navigation and Dead Reckoning: eCompass Tilt Compensation
- Robotics: Motion Sensing

| ORDERING INFORMATION | | | | |
|----------------------|-------------------|-----------------|---------|-----------------|
| Part Number | Temperature Range | Package Drawing | Package | Shipping |
| MMA7341LCT | -40 to +85°C | 1977-01 | LGA-14 | Tray |
| MMA7341LCR1 | -40 to +85°C | 1977-01 | LGA-14 | 7" Tape & Reel |
| MMA7341LCR2 | -40 to +85°C | 1977-01 | LGA-14 | 13" Tape & Reel |

MMA7341LC

MMA7341LC: XYZ AXIS ACCELEROMETER ±3g, ±9g

Bottom View



14 LEAD
LGA
CASE 1977-01

Top View

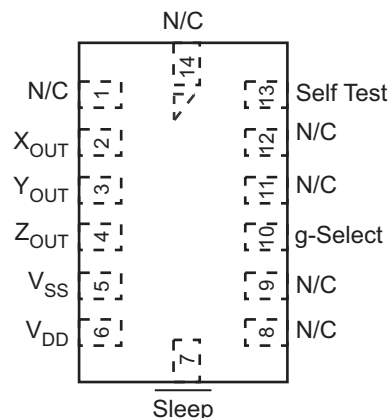


Figure 1. Pin Connections

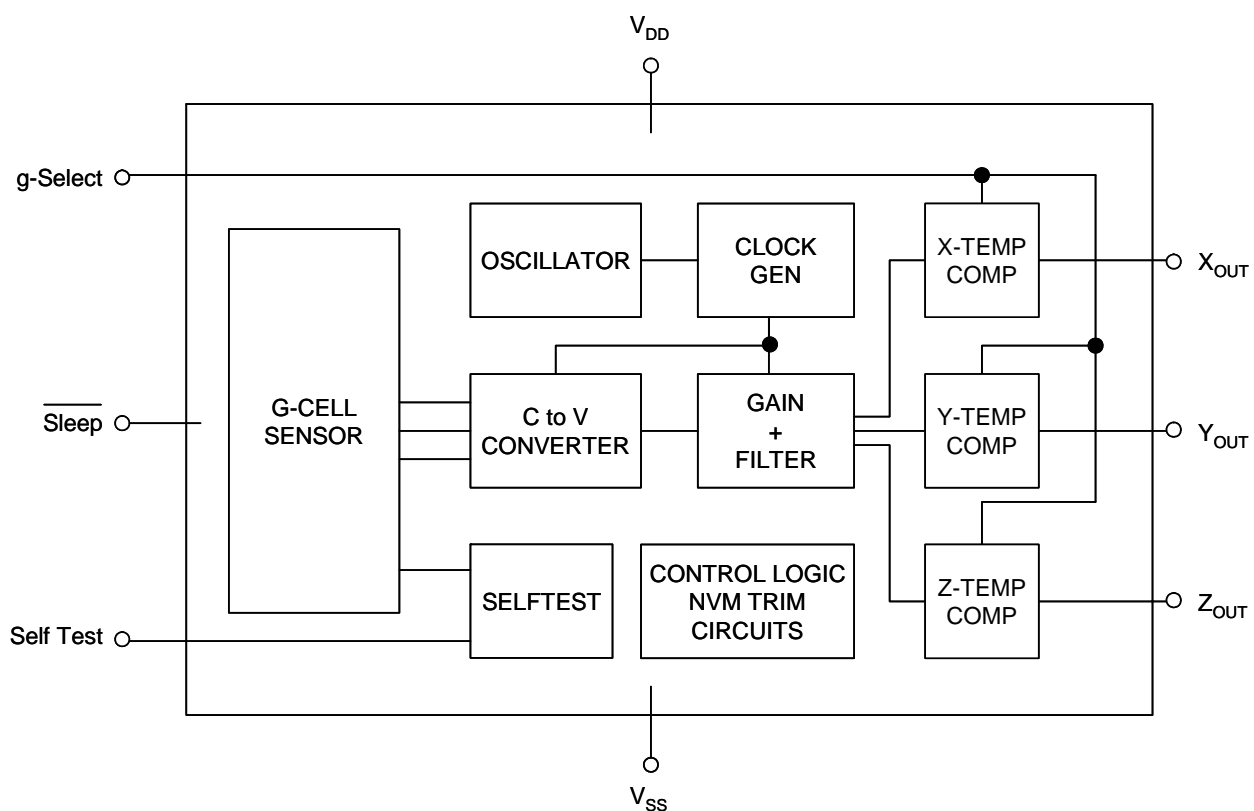


Figure 2. Simplified Accelerometer Functional Block Diagram

Table 1. Maximum Ratings

(Maximum ratings are the limits to which the device can be exposed without causing permanent damage.)

| Rating | Symbol | Value | Unit |
|---------------------------------|------------|--------------|------|
| Maximum Acceleration (all axis) | g_{max} | ± 5000 | g |
| Supply Voltage | V_{DD} | -0.3 to +3.6 | V |
| Drop Test ⁽¹⁾ | D_{drop} | 1.8 | m |
| Storage Temperature Range | T_{stg} | -40 to +125 | °C |

1. Dropped onto concrete surface from any axis.

ELECTRO STATIC DISCHARGE (ESD)

WARNING: This device is sensitive to electrostatic discharge.

Although the Freescale accelerometer contains internal 2000 V ESD protection circuitry, extra precaution must be taken by the user to protect the chip from ESD. A charge of over 2000 volts can accumulate on the human body or associated test equipment. A charge of this magnitude can

alter the performance or cause failure of the chip. When handling the accelerometer, proper ESD precautions should be followed to avoid exposing the device to discharges which may be detrimental to its performance.

Table 2. Operating CharacteristicsUnless otherwise noted: $-40^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$, $2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$, Acceleration = 0g, Loaded output⁽¹⁾

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---------------------|--------------|-------------|--------------|--------------------------------|
| Operating Range ⁽²⁾ | | | | | |
| Supply Voltage ⁽³⁾ | V_{DD} | 2.2 | 3.3 | 3.6 | V |
| Supply Current ⁽⁴⁾ | I_{DD} | — | 400 | 600 | μA |
| Supply Current at Sleep Mode ⁽⁴⁾ | I_{DD} | — | 3 | 10 | μA |
| Operating Temperature Range | T_A | -40 | — | +85 | $^{\circ}\text{C}$ |
| Acceleration Range, X-Axis, Y-Axis, Z-Axis | | | | | |
| g-Select: 0 | 9FS | — | ± 3 | — | g |
| g-Select: 1 | 9FS | — | ± 9 | — | g |
| Output Signal | | | | | |
| Zero g ($T_A = 25^{\circ}\text{C}$, $V_{DD} = 3.3\text{ V}$) ^{(5), (6)} | | | | | |
| XY | V_{OFF} | 1.551 | 1.65 | 1.749 | V |
| Z ⁽⁷⁾ | V_{OFF} | 1.23 | 1.65 | 1.749 | V |
| Zero g ⁽⁴⁾ | V_{OFF}, T_A | -2.0 | ± 0.5 | +2.0 | $\text{mg}/^{\circ}\text{C}$ |
| Sensitivity ($T_A = 25^{\circ}\text{C}$, $V_{DD} = 3.3\text{ V}$) | | | | | |
| 3g | S_{3g} | 413.6 | 440 | 466.4 | mV/g |
| 9g | S_{9g} | 106 | 117.8 | 129.6 | mV/g |
| Sensitivity ⁽⁴⁾ | S, T_A | -0.0075 | ± 0.002 | +0.0075 | $\%/^{\circ}\text{C}$ |
| Bandwidth Response | | | | | |
| XY | $f_{-3\text{dBXY}}$ | — | 400 | — | Hz |
| Z | $f_{-3\text{dBZ}}$ | — | 300 | — | Hz |
| Output Impedance | Z_O | — | 32 | — | $\text{k}\Omega$ |
| Self Test | | | | | |
| Output Response | | | | | |
| X_{OUT}, Y_{OUT} | Δg_{STXY} | +0.05 | -0.1 | — | g |
| Z_{OUT} | Δg_{STZ} | +1.0 | +2.5 | +4.0 | g |
| Input Low | V_{IL} | V_{SS} | — | $0.3 V_{DD}$ | V |
| Input High | V_{IH} | $0.7 V_{DD}$ | — | V_{DD} | V |
| Noise | | | | | |
| Power Spectral Density RMS (0.1 Hz – 1 kHz) ⁽⁴⁾ | n_{PSD} | — | 350 | — | $\mu\text{g}/\sqrt{\text{Hz}}$ |
| Control Timing | | | | | |
| Power-Up Response Time ⁽⁸⁾ | $t_{RESPONSE}$ | — | 1.0 | 2.0 | ms |
| Enable Response Time ⁽⁹⁾ | t_{ENABLE} | — | 0.5 | 2.0 | ms |
| Self Test Response Time ⁽¹⁰⁾ | t_{ST} | — | 2.0 | 5.0 | ms |
| Sensing Element Resonant Frequency | | | | | |
| XY | $f_{GCELLXY}$ | — | 6.0 | — | kHz |
| Z | f_{GCELLZ} | — | 3.4 | — | kHz |
| Internal Sampling Frequency | f_{CLK} | — | 11 | — | kHz |
| Output Stage Performance | | | | | |
| Full-Scale Output Range ($I_{OUT} = 3\text{ }\mu\text{A}$) | V_{FSO} | $V_{SS}+0.1$ | — | $V_{DD}-0.1$ | V |
| Nonlinearity, $X_{OUT}, Y_{OUT}, Z_{OUT}$ | NL_{OUT} | -1.0 | — | +1.0 | %FSO |
| Cross-Axis Sensitivity ⁽¹¹⁾ | $V_{XY, XZ, YZ}$ | -5.0 | — | +5.0 | % |

- For a loaded output, the measurements are observed after an RC filter consisting of an internal 32 k Ω resistor and an external 3.3 nF capacitor (recommended as a minimum to filter clock noise) on the analog output for each axis and a 0.1 μF capacitor on V_{DD} - GND. The output sensor bandwidth is determined by the Capacitor added on the output. $f = 1/2\pi * (32 \times 10^3) * C$. $C = 3.3\text{ nF}$ corresponds to $BW = 1507\text{ Hz}$, which is the minimum to filter out internal clock noise.
- These limits define the range of operation for which the part will meet specification.
- Within the supply range of 2.2 and 3.6 V, the device operates as a fully calibrated linear accelerometer. Beyond these supply limits the device may operate as a linear device but is not guaranteed to be in calibration.
- This value is measured with g-Select in 3g mode.
- The device can measure both + and – acceleration. With no input acceleration the output is at midsupply. For positive acceleration the output will increase above $V_{DD}/2$. For negative acceleration, the output will decrease below $V_{DD}/2$.
- For optimal 0g offset performance, adhere to AN3484 and AN3447.
- Product performance will not exceed this minimum level, however measurement over time will not be equal to time zero measurements for this specific parameter.
- The response time between 10% of full scale V_{DD} input voltage and 90% of the final operating output voltage.
- The response time between 10% of full scale Sleep Mode input voltage and 90% of the final operating output voltage.
- The response time between 10% of the full scale self test input voltage and 90% of the self test output voltage.
- A measure of the device's ability to reject an acceleration applied 90° from the true axis of sensitivity.

PRINCIPLE OF OPERATION

The Freescale accelerometer is a surface-micromachined integrated-circuit accelerometer.

The device consists of a surface micromachined capacitive sensing cell (g-cell) and a signal conditioning ASIC contained in a single package. The sensing element is sealed hermetically at the wafer level using a bulk micromachined cap wafer.

The g-cell is a mechanical structure formed from semiconductor materials (polysilicon) using semiconductor processes (masking and etching). It can be modeled as a set of beams attached to a movable central mass that move between fixed beams. The movable beams can be deflected from their rest position by subjecting the system to an acceleration (Figure 3).

As the beams attached to the central mass move, the distance from them to the fixed beams on one side will increase by the same amount that the distance to the fixed beams on the other side decreases. The change in distance is a measure of acceleration.

The g-cell beams form two back-to-back capacitors (Figure 3). As the center beam moves with acceleration, the distance between the beams changes and each capacitor's value will change, ($C = A\epsilon/D$). Where A is the area of the beam, ϵ is the dielectric constant, and D is the distance between the beams.

The ASIC uses switched capacitor techniques to measure the g-cell capacitors and extract the acceleration data from the difference between the two capacitors. The ASIC also signal conditions and filters (switched capacitor) the signal, providing a high level output voltage that is ratiometric and proportional to acceleration.

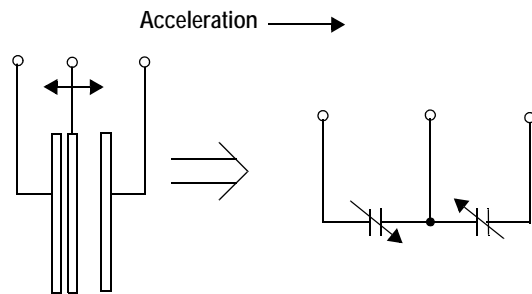


Figure 3. Simplified Transducer Physical Model

SPECIAL FEATURES

Self Test

The sensor provides a self test feature that allows the verification of the mechanical and electrical integrity of the accelerometer at any time before or after installation. This feature is critical in applications such as hard disk drive protection where system integrity must be ensured over the life of the product. Customers can use self test to verify the solderability to confirm that the part was mounted to the PCB correctly. When the self test function is initiated, an electrostatic force is applied to each axis to cause it to deflect. The X- and Y-axis are deflected slightly while the Z-axis is trimmed to deflect 1g. This procedure assures that both the mechanical (g-cell) and electronic sections of the accelerometer are functioning.

g-Select

The g-Select feature allows for the selection between two sensitivities. Depending on the logic input placed on pin 10, the device internal gain will be changed allowing it to function with a 3g or 9g sensitivity (Table 3). This feature is ideal when a product has applications requiring two different sensitivities for optimum performance. The sensitivity can be changed at anytime during the operation of the product. The g-Select pin can be left unconnected for applications requiring only a 3g sensitivity as the device has an internal pull-down to keep it at that sensitivity (440 mV/g).

Table 3. g-Select Pin Description

| g-Select | g-Range | Sensitivity |
|----------|---------|-------------|
| 0 | 3g | 440 mV/g |
| 1 | 9g | 117.8 mV/g |

Sleep Mode

The 3 axis accelerometer provides a Sleep Mode that is ideal for battery operated products. When Sleep Mode is active, the device outputs are turned off, providing significant reduction of operating current. A low input signal on pin 7 (Sleep Mode) will place the device in this mode and reduce the current to 3 μ A typ. For lower power consumption, it is recommended to set g-Select to 3g mode. By placing a high input signal on pin 7, the device will resume to normal mode of operation.

Filtering

The 3 axis accelerometer contains an onboard single-pole switched capacitor filter. Because the filter is realized using switched capacitor techniques, there is no requirement for external passive components (resistors and capacitors) to set the cut-off frequency.

Ratiometricity

Ratiometricity simply means the output offset voltage and sensitivity will scale linearly with applied supply voltage. That is, as supply voltage is increased, the sensitivity and offset increase linearly; as supply voltage decreases, offset and sensitivity decrease linearly. This is a key feature when interfacing to a microcontroller or an A/D converter because it provides system level cancellation of supply induced errors in the analog to digital conversion process.

BASIC CONNECTIONS

Pin Descriptions

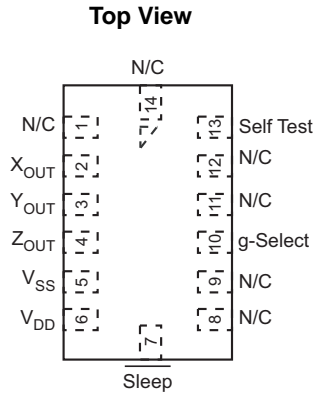


Figure 4. Pinout Description

Table 4. Pin Descriptions

| Pin No. | Pin Name | Description |
|---------|------------------|---|
| 1 | N/C | No internal connection Leave unconnected |
| 2 | X _{OUT} | X direction output voltage |
| 3 | Y _{OUT} | Y direction output voltage |
| 4 | Z _{OUT} | Z direction output voltage |
| 5 | V _{SS} | Power Supply Ground |
| 6 | V _{DD} | Power Supply Input |
| 7 | Sleep | Logic input pin to enable product or Sleep Mode |
| 8 | N/C | No internal connection Leave unconnected |
| 9 | N/C | No internal connection Leave unconnected |
| 10 | g-Select | Logic input pin to select g level |
| 11 | N/C | Unused for factory trim Leave unconnected |
| 12 | N/C | Unused for factory trim Leave unconnected |
| 13 | Self Test | Input pin to initiate Self Test |
| 14 | N/C | Unused for factory trim Leave unconnected |

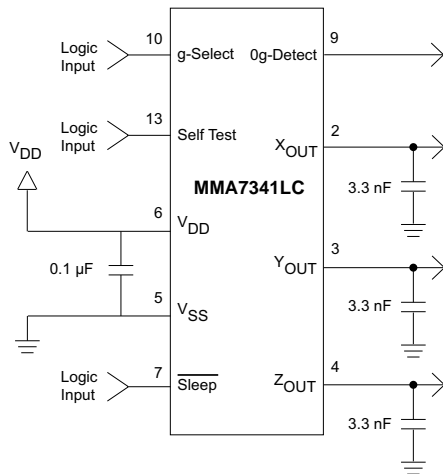


Figure 5. Accelerometer with Recommended Connection Diagram

PCB Layout

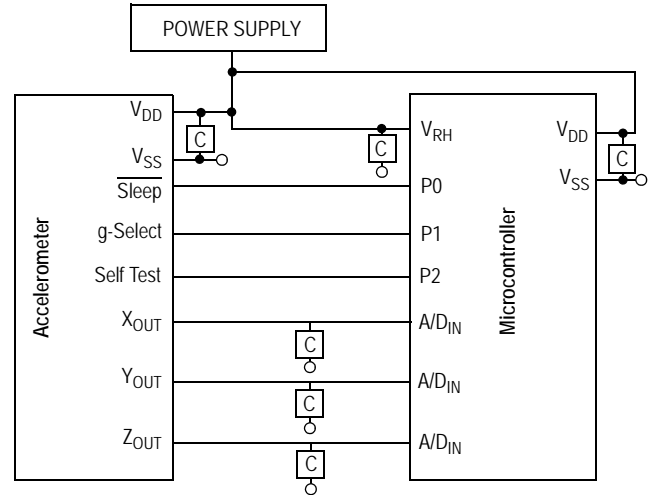
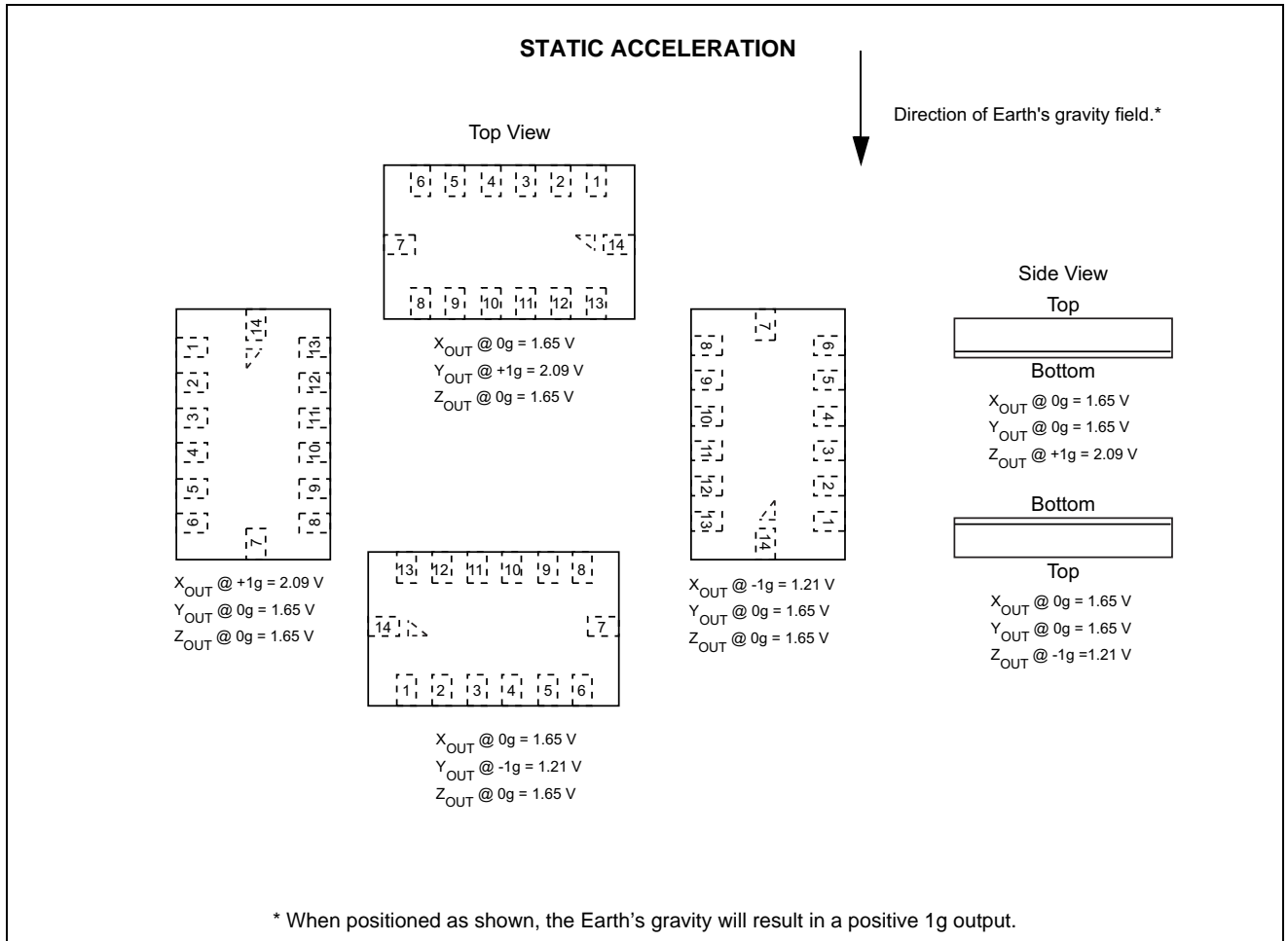
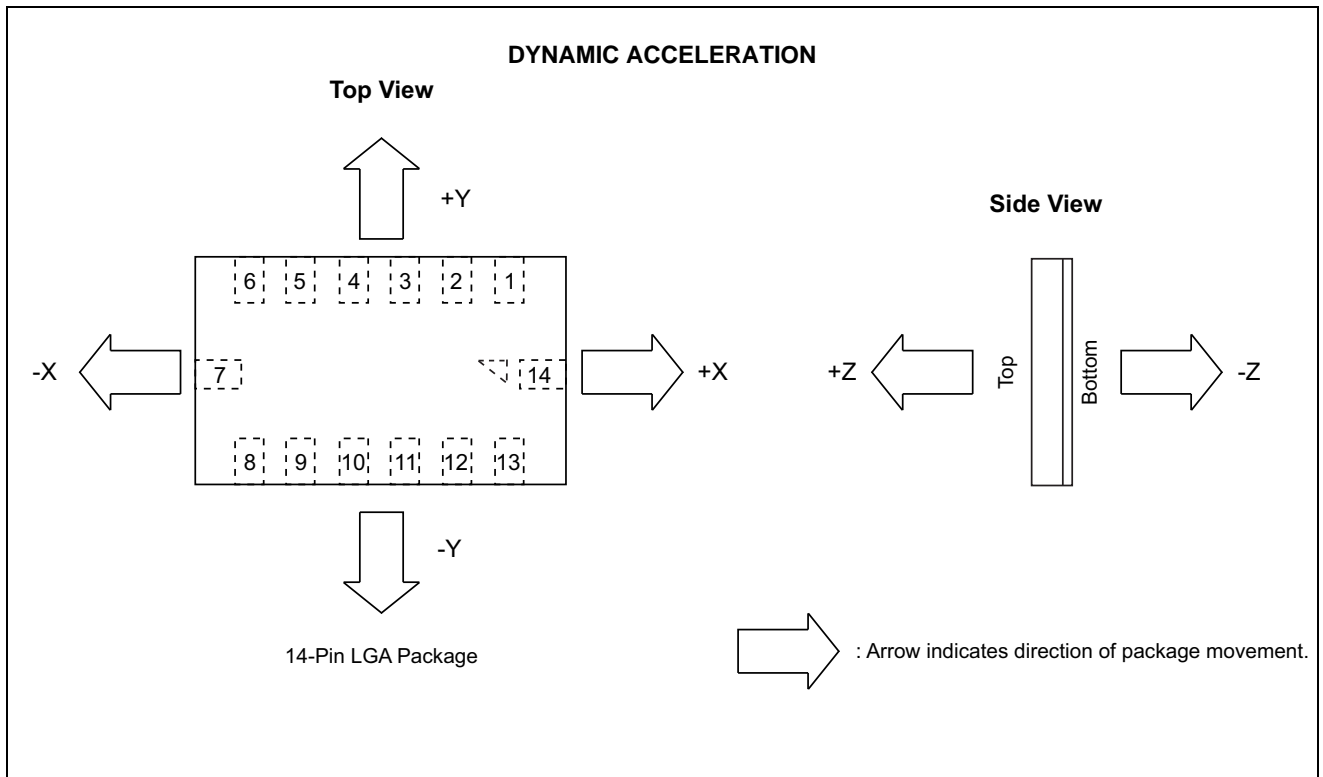


Figure 6. Recommended PCB Layout for Interfacing Accelerometer to Microcontroller

NOTES:

1. Use 0.1 μF capacitor on V_{DD} to decouple the power source.
2. Physical coupling distance of the accelerometer to the microcontroller should be minimal.
3. Place a ground plane beneath the accelerometer to reduce noise, the ground plane should be attached to all of the open ended terminals shown in [Figure 6](#).
4. Use a 3.3 nF capacitor on the outputs of the accelerometer to minimize clock noise (from the switched capacitor filter circuit).
5. PCB layout of power and ground should not couple power supply noise.
6. Accelerometer and microcontroller should not be a high current path.
7. A/D sampling rate and any external power supply switching frequency should be selected such that they do not interfere with the internal accelerometer sampling frequency (11 kHz for the sampling frequency). This will prevent aliasing errors.
8. 10 M Ω or higher is recommended on X_{OUT}, Y_{OUT} and Z_{OUT} to prevent loss due to the voltage divider relationship between the internal 32 k Ω resistor and the measurement input impedance.



* When positioned as shown, the Earth's gravity will result in a positive 1g output.

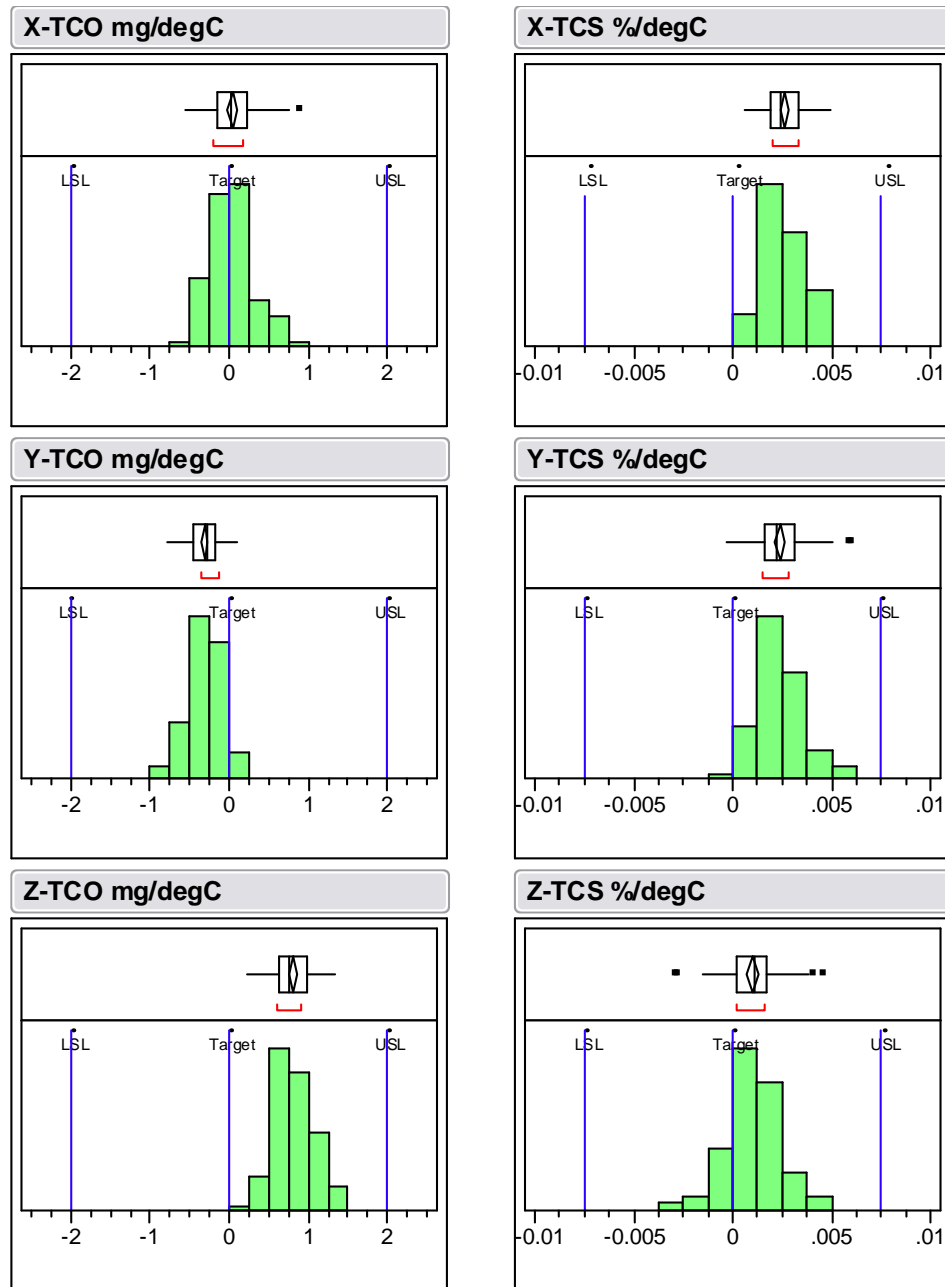


Figure 7. MMA7341LC Temperature Coefficient of Offset (TCO) and Temperature Coefficient of Sensitivity (TCS) Distribution Charts

MINIMUM RECOMMENDED FOOTPRINT FOR SURFACE MOUNTED APPLICATIONS

PCB Mounting Recommendations

MEMS based sensors are sensitive to Printed Circuit Board (PCB) reflow processes. For optimal zero-g offset after PCB mounting, care must be taken to PCB layout and reflow conditions. Reference application note AN3484 for best practices to minimize the zero-g offset shift after PCB mounting.

Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package.

With the correct footprint, the packages will self-align when subjected to a solder reflow process. It is always recommended to design boards with a solder mask layer to avoid bridging and shorting between solder pads.

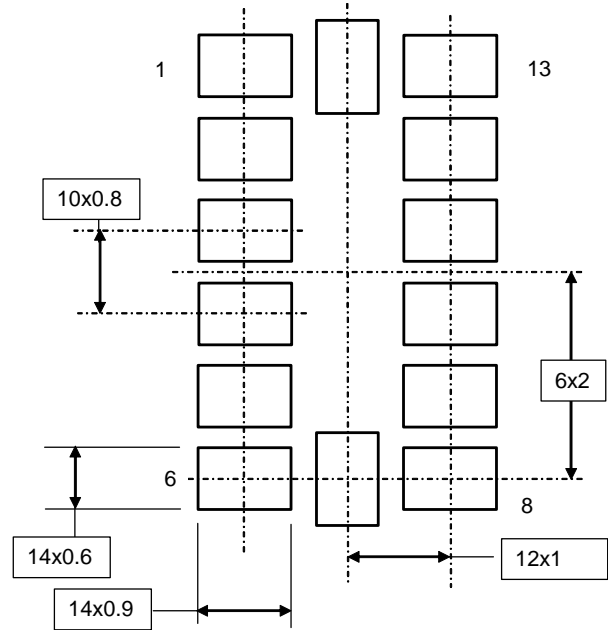
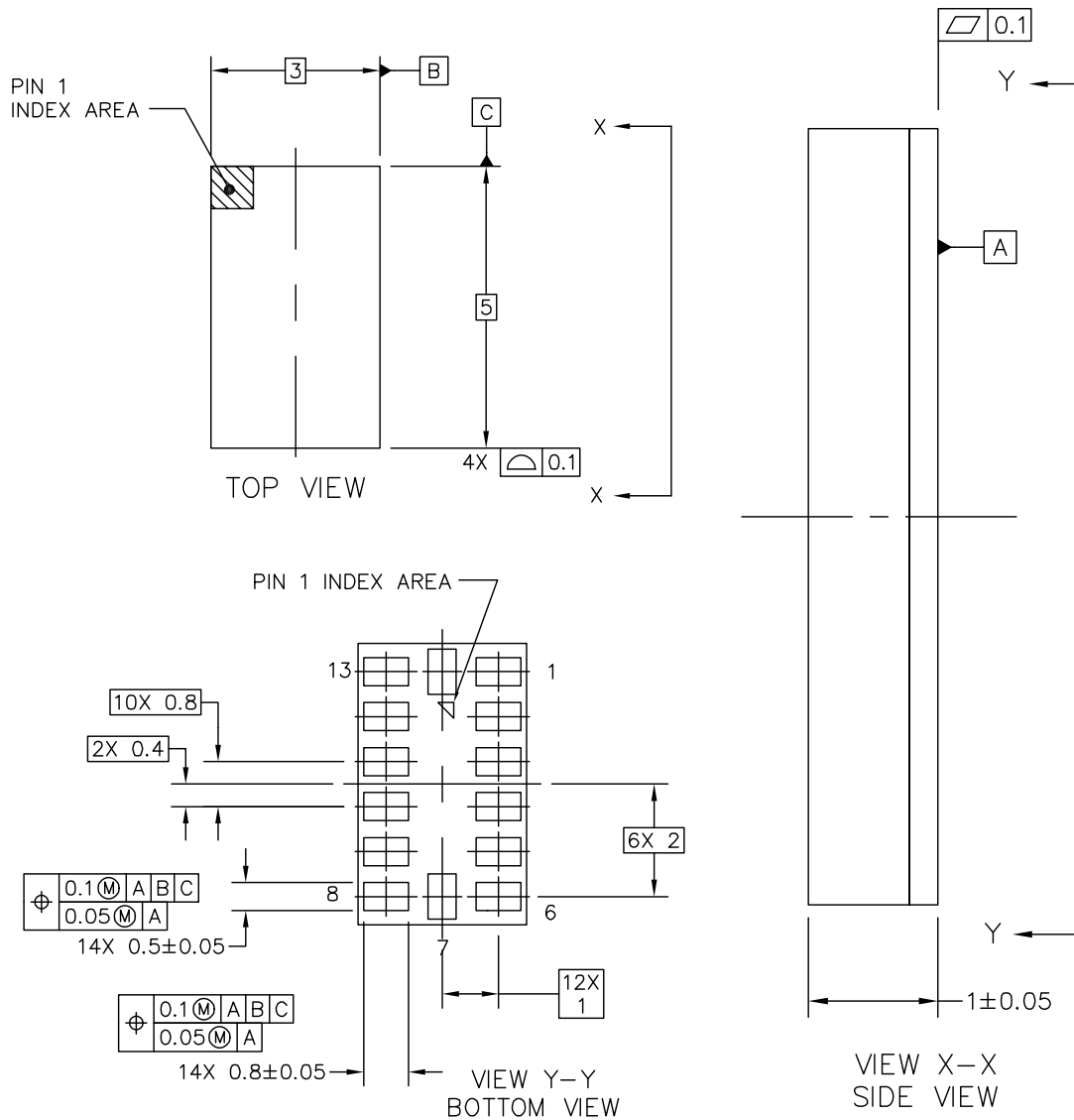


Figure 8. LGA 14-Lead, 5 x 3 mm Die Sensor

PACKAGE DIMENSIONS



| | | | |
|---|---------------------------|----------------------------|-------------|
| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE | |
| TITLE: LGA 14 I/O, 3 X 5 X 1 PITCH 0.8, SENSOR 1.0MM PKG | DOCUMENT NO: 98ASA10801D | | REV: A |
| | CASE NUMBER: 1977-01 | | 09 JAN 2008 |
| | STANDARD: NON-JEDEC | | |

**CASE 1977-01
ISSUE A
14-LEAD LGA**

MMA7341LC

PACKAGE DIMENSIONS

NOTES:

1. ALL DIMENSIONS IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.

| | | | | |
|---|--------------------------|--|----------------------------|--|
| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | | PRINT VERSION NOT TO SCALE | |
| TITLE: LGA 14 I/O, 3 X 5 X 1 PITCH 0.8, SENSOR 1.0MM PKG | DOCUMENT NO: 98ASA10801D | | REV: A | |
| | CASE NUMBER: 1977-01 | | 09 JAN 2008 | |
| | STANDARD: NON-JEDEC | | | |

**CASE 1977-01
ISSUE A
14-LEAD LGA**

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale and the Freescale logo are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. Xtrinsic is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2011. All rights reserved.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Annex III. Arduino Due code to read the output of the sensors and generate the PWM signal

```

//PWM
#define PWM_INTERFACE      PWM
#define PWM_INTERFACE_ID  ID_PWM
#define PWM_FREQUENCY      1000
#define PWM_MAX_DUTY_CYCLE  4095  // changed from 255
#define PWM_MIN_DUTY_CYCLE  0
#define PWM_RESOLUTION      12    // changed from 8
//

#include <DueTimer.h>

#define COMMAND_LENGTH 100
#define NELEMENTS 100

// INITIALIZING ALL VARIABLES
char command[COMMAND_LENGTH];
int sIt;
char c;
int numel;
int count = 0;
int ref_data = 0;

float v[NELEMENTS];
float va;
int it = 0;
float vm = 0;
float sc = 5.0/4096.0; // To scale the voltage output

int pwmPin = 7;
int signalPin = A0;

boolean runProgram = false;
boolean puja = true;

unsigned long t = 0 ;

int PWM_Val = 50;

// Controller
//float Kp = 1.62e3, Ki = 1.53e3; // Matlab Values
float Kp = 0.1, Ki = 1.1;      // Tested values

float T = 0.01; // Period
float error = 0, error_1 = 0, Control_Signal = 0, Control_Signal_1 = 0, Control_Signal_Sat=0;

float ref = 50; // Reference value we want to achieve - it can be
changed by using the 'set' function

// Saturation
int umax = 1050; // Maximum duty
int umin = 900;  // Minimum duty

// Anti-windup
float w = 0, wAnt = 0;
float Kw = Ki/Kp;

void setup()
{
    Serial.begin(115200);

```



```

pinMode(pwmPin, OUTPUT);
pinMode(signalPin, INPUT);

Timer1.attachInterrupt(upFun).setPeriod(20200).start(); //20e3
Timer2.attachInterrupt(downFun);
Timer3.attachInterrupt(Obtain_Data);
Timer4.attachInterrupt(AverageVolts).setPeriod(1000).start();
//1000
}

void loop()
{
    // put your main code here, to run repeatedly:
}

// Function to calculate the average of the sensor's output
void AverageVolts()
{
    va = (float)analogRead(signalPin);/*sc ;
    vm = vm + (va - v[it])/(NELEMENTS);
    v[it] = va;
    it++;
    if(it==NELEMENTS) it = 0;
}

// Function to print: Time, sensor output, data
void PrintData()
{
    Serial.print(micros());
    Serial.print(" ");
    Serial.print(PWM_Val);
    Serial.print(" ");
    Serial.println(ref_data-vm);
}

// Function used to obtain the sensor's output wby increasing and
decreasing the duty
void Obtain_Data()
{
    if(runProgram == true)
    {
        if((millis()-t) >= 5e3)
        {
            t = millis();
            if(puja == true)
            {
                PWM_Val = PWM_Val + 10;
                if(PWM_Val >= 1400) // 1400 - Maximum duty that can be
set, otherwise the bar is in the unstable part
                {
                    puja = false;
                }
            }
            else
            {
                PWM_Val = PWM_Val - 20;

```

```

        if(PWM_Val <= 870)          // 870 - Maximum duty to finally
stop the ESC
        {
            Timer3.stop();
        }
    }

    PrintData();
}
else
{
    PWM_Val = PWM_Val - 5;
    if(PWM_Val <= 700) Timer3.stop();
}
}

// Controller Function
void Controller()
{
    error = ref - (ref_data-vm);          //error
    w = Control_Signal - Control_Signal_Sat; //anti-windup

    Control_Signal = error * Kp +
        (-Kp + Ki * T) * error_1 +
        Control_Signal_1 - Kw*T/2 * (w+wAnt);

    Control_Signal_1 = Control_Signal;
    error_1 = error;
    wAnt = w;

    // Saturator
    Control_Signal_Sat = Control_Signal;
    if(Control_Signal_Sat > umax) Control_Signal_Sat = umax; //1020
    if(Control_Signal_Sat < umin) Control_Signal_Sat = umin; //960

    PWM_Val = Control_Signal_Sat;
    call_Controller();
}

// This function is printing the different values realted with the
controller.
// Depending on the needs it can be commented or decommended to
print the data on the monitor
void call_Controller()
{
    Serial.print(micros());
    Serial.print(" ");
    Serial.print(PWM_Val);
    Serial.print(" ");
    Serial.println(ref_data-vm);
    //Serial.print(" ");
    //Serial.println(error);
    //Serial.print(" ");
    //Serial.print("PWM: ");
    //Serial.println(PWM_Val);
    //Serial.print("Value dist: ");
    //Serial.println(ref_data-vm);
    //Serial.print("ERROR: ");

```

```

    //Serial.println(error);
}

// Function to reset the variables in case it is needed
void reset()
{
    Control_Signal_1 = 0;
    error = 0;
    error_1 = 0;
    Serial.println("Reset");
}

// Function controlling the duty by setting the PWM pin - it is
called every 2ms
void upFun()
{
    digitalWrite(pwmPin, HIGH);
    Timer2.setPeriod(PWM_Val).start();
}

// Function to stop the PWM power
void downFun()
{
    digitalWrite(pwmPin, LOW);
    Timer2.stop();
}

void serialEvent() {
    while (Serial.available())
    {
        c = Serial.read();
        if (c == 13)
        {
            command[sIt] = 0; //null (end of array)
            numel = sIt-1;
            sIt = 0;
            analizeString();
        }
        else
        {
            command[sIt] = c;
            sIt++;
            if (sIt == COMMAND_LENGTH) sIt = 0;
        }
    }
}

// Different funcions created for performing tests and controlling
the motor
void analizeString() {
    int m;
    switch (command[0]) {
        case 'v': // Control manually the duty given to the ESC
            PWM_Val = atoi(&command[1]);
            Serial.println(PWM_Val);
            Timer3.stop();
            Timer3.setPeriod(T*1e6);

```

```

        Timer3.attachInterrupt(PrintData);
        Timer3.start();
        break;
    case 't': // Call Test mode - getting data automated test by
increasing the duty constantly
        puja = true;
        PWM_Val = 870;
        runProgram = true;
        t = millis();
        Timer3.setPeriod(0.01e6).start();
        break;
    case 'p': // Stop the test program slowly
        runProgram = false;
        break;
    case 'r': // Calculate the Reference value
        for(m=0; m<1000; m++) {AverageVolts(); delay(5);}
        ref_data = vm;
        Serial.println(ref_data);
        break;
    case 'c': // Call the controller function
        for(m=0; m<100; m++) {AverageVolts(); delay(5);}
        ref_data = vm;
        Timer3.stop();
        Timer3.setPeriod(T*1e6);
        Timer3.attachInterrupt(Controller);
        Timer3.start();
        break;
    case 'u': // Reset controller parameters
        reset();
        break;
    case 's': // Set a different reference point for the controller
        ref = atoi(&command[1]);
        break;
    }
    Serial.print("La comanda es: ");
    Serial.println(command);
}

```

Annex IV. Arduino Mega code to read the output of the sensors and generate the PWM signal

```

#include <TimerOne.h>
#define NELEMENTS 200
#define COMMAND_LENGTH 100

char command[COMMAND_LENGTH];
int sIt;
char c;
int numel;
int count = 0;

float v[NELEMENTS];
float va;
int it = 0;
float vm = 0;
float sc = 5.0/1023.0;

unsigned long time ;
float t = 0;

int pwmPin = 11;
int signalPin = A0;

int power = 64;

boolean RunProgram = false;
boolean puja = true;

//unsigned long t = 0;

void setup() {
    Serial.begin(2e6);
    pinMode(pwmPin, OUTPUT);
    pinMode(signalPin, INPUT);
    Timer1.initialize(20000);           // initialize timer1, and set a
1/2 second period
    Timer1.pwm(pwmPin, power);
    Timer1.attachInterrupt(Funcio_Timer); // attaches Funcio_Timer()
as a timer overflow interrupt

}

void loop() {
    EstimaVolts();
}

void EstimaVolts()
{
    va = (float)analogRead(signalPin);/*sc ;
    vm = vm + (va - v[it])/(NELEMENTS);
    v[it] = va;
    it++;
    if(it==NELEMENTS) it = 0;
}

void Funcio_Timer()
{
    if(RunProgram == true)
    {
        if((millis()-t) >= 2e3)
        {

```

```

        t = millis();
        if(puja == true)
        {
            power = power + 1;
            Timer1.pwm(pwmPin, power);
            if(power >= 74)
            {
                puja = false;
            }
        }
        else
        {
            power = power - 1;
            Timer1.pwm(pwmPin, power);
            //Serial.println(power);
            if(power == 65)
            {
                RunProgram = false;
            }
        }
    }
    Serial.print(micros());
    Serial.print(" ");
    Serial.print(power);
    Serial.print(" ");
    Serial.println(vm);
}

}

void serialEvent() {
    while(Serial.available())
    {
        c = Serial.read();
        if(c == 13)
        {
            command[sIt] = 0; //null (end of array)
            numel = sIt-1;
            sIt = 0;
            analizeString();
        }
        else
        {
            command[sIt] = c;
            sIt++;
            if(sIt == COMMAND_LENGTH) sIt = 0;
        }
    }
}

void analizeString(){
    switch (command[0])
    {
        case 'v':
            power = atoi(&command[1]);
            Timer1.pwm(pwmPin, power);
            Serial.println(power);
            break;
        case 't':
            RunProgram = true;
    }
}

```

```
        t = millis();  
        break;  
    }  
  
    Serial.print("La comanda es: ");  
    Serial.println(command);  
  
}
```


Annex V. Datasheet of the brushless motor

| | |
|------------------|--------------|
| SKU | 9192000151-0 |
| Shipping Weight | 82.0000 |
| Packaging Width | 50.00 |
| Kv(rpm/v) | 2150.00 |
| Resistance (mh) | 0.00 |
| Power (W) | 120.00 |
| Length B (mm) | 17.00 |
| Can Length D(mm) | 7.00 |

| | |
|--------------------|-----------|
| Brand | MultiStar |
| Packaging Length | 100.00 |
| Packaging Height | 80.00 |
| Max Current (A) | 10.00 |
| Max Voltage(V) | 12.00 |
| Shaft A(mm) | 5.00 |
| Can Diameter C(mm) | 28.00 |
| Total Length E(mm) | 34.00 |

Annex VI. Datasheet of the Afro ESC 30A



AfroESC 30A USER MANUAL

Thank you for purchasing AfroESC, with a high efficiency, all N-channel MOSFET design, jitter-free hardware PWM input, and smooth power response! The AfroESC series ship with SimonK software, suitable for multi-rotor use without the need to program or adjust settings.

Caution

With any new build or after making any changes, always test carefully in a controlled environment. Ensure propellers rotate in the correct direction and are tightened, that all signal connections are in the right order and firmly connected. Motors connected in the wrong order may result in uncontrolled flips. Never attempt to stop propellers by hand, and maintain a safe distance at all times.

The AfroESC 30A has a lightweight BEC, capable of about 0.5A, which should be sufficient to power most flight boards, but not servos. A dedicated BEC is recommended where servos or other devices are used.

There is NO protection against reverse polarity, which would reduce efficiency and add weight. Be sure that you connect RED to POSITIVE (+V) and BLACK to GROUND when wiring or attaching bullets to avoid damage.

Software and Updates

If desired, the firmware may be upgraded and/or customized to enable or disable various features such as complementary PWM, brake, RC-Car mode, or adjust default settings such as endpoints and motor timing. The firmware can be updated using similar procedures as for the KK(2) and other projects based on ATmega hardware. For source code, binaries, and other information, see <http://github.com/sim-tgy/> and <http://0x.ca/tgy/>.

The easiest way of updating is with the Afro USB Linker from HobbyKing attached to the regular PWM input cable. Only the signal and ground connections are required. The boot loader speaks STK500v2 at 9600 baud using the half duplex protocol implemented by the Afro USB Linker. Other linkers (such as USB serial linkers) will not work at this time.

The KKMulticopter Flash Tool or any general AVR ISP software may be used with any of the above options. See the above URLs for links and more information.

Default Settings

Input control is available through PWM, or through I²C (MK) or UART with additional wiring (auto-detected).

For PWM input, motor stop / arm below 1060μs, full power at 1860μs.

For I²C input, motor stop / arm at 0, start at 1, full power at 247. BL-Ctrl V2 reverse supported.

For UART input, motor stop / arm at 0, start at 1, full power at 200.

Low-voltage cut-off (input voltage checking) and temperature limiting are not enabled, as reacting to such conditions on a multi-rotor will likely be worse than ignoring them. Always use a LiPo battery alarm or telemetry to avoid over-discharging LiPo batteries!

General Operation

Once an input signal is received that is low enough to reach the “neutral” or power-off area, the ESC will arm (long beep), and the green LED will light. Once the signal enters the motor start range, the ESC will attempt to start or run the motor, and continue to do so as long as input signal is above the neutral position. The amount of power used for starting is not fixed, just based on the power requested, limited and ramped at first to increase the chances of a successful start.

In general, the power handling should be as smooth as possible, but not limited or filtered once running. The default PWM frequency is 18kHz with 800 distinct steps, but may be adjusted to any frequency. During starting, the PWM frequency is halved both to assist starting of some difficult motors and as an audible indication of this state.

There is no latching of stopped or start states, as doing so could make acrobatic manoeuvres unsafe. It should not ever be required to re-arm the ESC unless some signal, power, or hardware fault occurs. In the event that an object (such as a tree) is struck by a moving propeller, the motor will limit the power output until timing is regained, but not require re-arming. In the event that the propeller is workable, this may permit a safer landing.

If a commutation time-out or long demagnetization period is detected, the red LED will light. This should not occur in normal operation. Very brief flashes during rapid acceleration indicate that the demagnetization period exceeded the expected zero-crossing point and that countermeasures have been taken to avoid loss of synchronization. The green LED will remain off once running until the power is turned off and the motor stops.

Throttle Range Calibration

With normal PWM input, ESCs and flight boards should be configured to match the expected working range. If unmatched, the motors may not start until higher throttle signal is given, may start “hot” even with a low throttle, may reach full power early, or may not reach full power. In many cases, end-point calibration is not required unless the flight board outputs different ranges.

First, remove all propellers! Check to see if there are calibration steps documented for your flight controller board. KK2 boards have a two-button procedure that performs automatical calibration. With KK boards, the Yaw pot must be set to the minimum position in order to enable pass-through mode from the receiver input to the ESC output, and the ESCs are then calibrated to the radio's throttle throw. With the Naze32, try the console settings “set minthrottle=1064” and “set maxthrottle=1864”. The same settings may be used for AutoQuad or other boards that allow the endpoints to be specified.

If none of the above options are available, calibration may be performed manually. First, remove all propellers! Disconnect the power to the ESCs. Connect the ESC PWM input directly to the receiver's throttle channel, or to a servo tester.

Set the radio throttle or servo tester to the highest position, then connect power to the ESC. The motor should produce a series of initialization beeps increasing in pitch, followed by another beep matching the pitch of the last initialization beep. This indicates that the calibration mode has been entered, and the pulse length has been learned.

Move the stick or knob to the lowest position. Two beeps of the same pitch should be emitted. This indicates that the low pulse length has been learned. If the RC Car-style reversible mode has been enabled (RC_PULS_NEUTRAL), move the stick or knob to the center, and wait for three beeps. This indicates that the neutral (center) pulse length has been learned.

The ESC will then save the settings and exit calibration mode. If the input is still at the same position, the ESC will arm (producing a higher pitched, long beep), and function normally.

NOTE: Throttle calibration is disabled to avoid accidental calibration if the ESC is reset by brown-out during previous operation. If the power is not connected cleanly during the above steps, the power-up may also be detected as a brown-out. If the rising beeps are not heard while attempting to calibrate, disconnect the power for a few seconds, then try again.

Troubleshooting

To aid with troubleshooting, the AfroESC sports “ready” (green) and “error” (red) LEDs. During power-up, the MOSFETs and drive circuitry are checked to ensure correct operation. During this test, it may be unsafe to beep an error. If no beeps are heard from the motor, check to see if the error LED is flashing, which should be visible through the heat-shrink. If so, count the number of blinks between pauses and refer to the following list:

- 1 flash: Phase A stuck high
- 2 flashes: Phase B stuck high
- 4 flashes: Phase C stuck high
- 5 flashes: AIN0 (center neutral) stuck high
- 6 flashes: Phase A low-side drive fault
- 7 flashes: Phase B low-side drive fault
- 8 flashes: Phase C low-side drive fault
- 9 flashes: Phase A high-side drive fault
- 10 flashes: Phase B high-side drive fault
- 11 flashes: Phase C high-side drive fault

If no wiring fault or short is visible, one or more MOSFETs may have failed. The ESC must be repaired or replaced. Otherwise, if the hardware check passes, the reason for reset is checked and

indicated by beeping through the motor and LED flashing. There are four main beep pitches:

Beep 1: Lowest pitch (also red LED).

Beep 2: Medium-low pitch (also green LED).

Beep 3: Medium-high pitch (no LEDs)

Beep 4: Highest pitch (also red and green LEDs together).

Beeps 1, 2, and 3: Normal power-up with no special event detected. A longer beep 4 normally follows to indicate that an arming signal was detected.

Beep 3, beep 1: Voltage brown-out detected (MCU voltage dropped below 4.0V). If this happens during battery connection, it can be safely ignored. If this happens during use, the ESC may have drawn more than the power source is able to provide, and the input voltage dropped below about 5V.

Beep 4: External reset. Should only occur after programming.

Repeating beeps 1, 1, 3, 3: Watchdog siren. Previous execution locked up or was not able to signal proper status, so to prevent motor or other damage, the ESC reset itself. Check for moisture problems or strong fields right next to the ESC. If nothing is found, the MCU may be faulty.

Once a valid input is connected and the ESC sees an idle power request, a long beep 4 is played and the green LED lit to indicate that the ESC is armed. This is required for all input sources, and must be maintained at a rate of at least 20Hz, or power will be switched off. If the input signal stops, the ESC will disarm after 2 seconds and play beeps 3 and 2 to indicate that the input disappeared. The ESC will look again for valid input sources, or programming input via the Turnigy USB linker.

If no input is detected for more than 8 seconds, a periodic beep 3 every 3 seconds will be emitted (beacon mode). This may aid in the location of a crash, assuming the signal is lost.